

Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería de Telecomunicación



Proyecto Fin de Carrera

**“Diseño y Desarrollo de un Marco para
Aplicaciones de Difusión Selectiva de
Contenido Multimedia”**

Autor: David Timón Morillo-Velarde

Tutora: Dra. María Calderón Pastor

Director: Dr. Ignacio Soto Campos

Mayo de 2011

PROYECTO FIN DE CARRERA

Departamento de Ingeniería Telemática

Universidad Carlos III de Madrid

Título: Diseño y Desarrollo de un Marco para Aplicaciones de Difusión Selectiva de Contenido Multimedia

Autor: David Timón Morillo-Velarde

Tutora: Dra. María Calderón Pastor

Director: Dr. Ignacio Soto Campos

La lectura y defensa del presente proyecto final de carrera se realizó el día 13 de mayo de 2011 bajo el tribunal:

- **Presidente:** Dr. Antonio de la Oliva Delgado
- **Secretario:** Dra. Iria Manuela Estévez Ayres
- **Vocal:** Dr. Ángel Bravo Santos

Habiendo obtenido la calificación de:

Presidente

Secretario

Vocal

*"Es necesario aprender lo que necesitamos
y no únicamente lo que queremos."*

Paulo Coelho

*"Lo que tenemos que aprender,
lo aprendemos haciéndolo."*

Aristóteles

AGRADECIMIENTOS

En primer lugar reconocer y agradecer al Dr. Ignacio Soto su labor como director, tanto por su conocimiento y sugerencias aportadas, siempre acertadas, como por su calidad humana y comprensiva: él siempre ha sido una de las personas que más se han preocupado e interesado por la finalización de este trabajo y el comienzo de una nueva etapa en mi vida.

Agradecerle a la Dra. María Calderón su disponibilidad en todo momento, ella ha sido la mejor tutora que se podría haber tenido.

Agradecer a la *Universidad Carlos III de Madrid*, en general, la oportunidad que me ha brindado de poder superar una etapa en la que, además de los conocimientos técnicos que me ha aportado, me ha inyectado una dosis importante de emprendimiento y me ha inculcado ciertos valores que, aunque no aparecen en el plan de estudio, considero de un grandísimo valor.

En particular agradecer a Cassia Silva y Emma Crespo su especial contribución a mi formación desde el *Vivero de Empresas de la Universidad Carlos III de Madrid*. Aunque nuestra relación aparezca lejana en el tiempo, considero que ha sido una importante aportación en mi educación, ya que me ha permitido entender la ingeniería desde otros puntos de vista, por ello, me considero un privilegiado.

Muchas gracias a todos los profesores que con su entusiasmo han sabido transmitir, a mi entender, el objetivo final de qué es convertirse en un ingeniero. Opino que es básicamente resolver problemas utilizando el conocimiento especializado, donde la imaginación y la innovación, muchas veces, marcan la diferencia. De corazón, gracias a todos.

A mi familia, por sus muestras de interés y preocupación, los cuales siempre han sabido aconsejarme sabiamente y me han apoyado sin condiciones en los momentos más duros, papá, mamá, hermanos, abuelas, tíos, primos... muchísimas gracias.

A mis compañeros y, sobre todo, amigos de la universidad; fundamentalmente a los tres sectores: el extremeño, albaceteño y madrileño. Nuestra aventura comenzó con la incansable lucha por mejorar, muchas gracias Javi y Juan, Diana, Juan Carlos, Salva, 'Juanjos', Ana, Javi y Cantero, Paloma... habéis convertido situaciones difíciles en momentos inolvidables, sobre todo al mantener siempre un espíritu positivo y hacer evidencia de ese inteligente sentido del humor que os acompaña: habéis sido los mejores en los peores momentos.

A mis amigos de siempre, los cuales han mostrado interés a lo largo de estos años, gracias Ángel, Paco, Joaquín, Borja, Alberto, Rubén, Curri, Pelu, Martín, Kiko... sois muy buenos amigos y os estoy muy agradecido: chicos, muchas gracias.

Y a Esti que, aunque ha sido la última en incorporarse a esta etapa de mi vida, se ha convertido en un gran apoyo. Ha sido muy importante para mí vislumbrar cada muestra de comprensión que me ha ofrecido. Muchas gracias por tu apoyo y cariño.

ÍNDICE

PARTE I. INTRODUCCIÓN.....	19
CAPÍTULO 1 INTRODUCCIÓN	19
1.1 Objetivos del Proyecto	20
1.2 Estructura de la Memoria	21
PARTE II. ESTADO DEL ARTE. SOLUCIONES COMERCIALES.....	24
CAPÍTULO 2 CONCEPTOS Y CONSIDERACIONES DE DIFUSIÓN SELECTIVA	24
2.1 Definición Difusión Selectiva o Narrowcasting	24
2.1.1 Cartelería Digital (Digital Signage) frente a la Difusión Selectiva (Narrowcasting)	25
2.1.2 Servicios y Aplicaciones en la Sociedad de la Información	25
2.2 Tipos de Enlaces de Comunicación Básicos	28
2.3 Difusión Selectiva (Narrowcast) y Comunicaciones (Unicast, Broadcast y Multicast)	28
2.4 Tecnologías de Red para Transportar Tráfico de Aplicaciones de Difusión Selectiva.....	31
2.4.1 Redes de Distribución de Área Amplia (WAN)	31
2.4.2 Redes de Distribución de Área Local (LAN).....	33
2.5 Definición de Tipos de Difusión Selectiva	34
2.5.1 Difusión Selectiva Pasiva	34
2.5.2 Difusión Selectiva Adaptativa o Activa	35
2.5.3 Difusión Selectiva Interactiva	36
2.6 Transporte del Contenido Multimedia en Difusión Selectiva.....	36
2.6.1 Descarga y Reproducción, Pseudo Streaming (Descarga Progresiva) y Streaming Media	37
2.6.2 Alternativas al Transporte del Contenido Multimedia	38
2.6.3 Codificación del Contenido Multimedia. Contenedores Multimedia.	40
2.7 Resumen	45
CAPÍTULO 3 SOLUCIONES PARA PROPORCIONAR SERVICIOS DE DIFUSIÓN SELECTIVA	47
3.1 Introducción a las Soluciones de Difusión Selectiva	47
3.2 Características de las Soluciones Encontradas	50
3.2.1 Aplicaciones Software de Propósito Específico y aplicaciones de Carácter General	51
3.2.2 Arquitecturas Cliente-Servidor y Arquitecturas Distribuidas	52
3.2.3 Arquitecturas Auto-Proxy o Bajo Demanda.....	53
3.2.4 Arquitecturas Basadas en Software de Aplicación y Arquitecturas Basado en herramientas	
Web	54
3.2.5 Arquitecturas con Hardware Genérico, Hardware Específico y Hardware Integrado o	
Embebido	55
3.3 Evolución Detectada de las Arquitecturas de Difusión Selectiva	56
3.4 Soluciones de Difusión Selectiva	57
3.4.1 Solución Propietaria de Referencia: Cisco Digital Media System (Suite)	58
3.4.2 Aplicaciones de la Solución de Referencia.....	61
3.4.2.1 Cartelería Digital (Digital Signage)	61
3.4.2.2 Portal (Web) de Video	62
3.4.2.3 Sistema de TV Corporativo (Enterprise TV)	63
3.4.3 Arquitectura Funcional de Referencia.....	64
3.4.3.1 Clasificación de los Elementos dentro de la Arquitectura de Referencia	66
3.4.3.2 Espacio de Funcionamiento Simple	67
3.4.3.3 Espacio de Funcionamiento Distribuido.....	69
3.4.3.4 Espacio de Funcionamiento Híbrido	74
3.5 Arquitectura de Difusión Selectiva de Referencia Completa	76
3.6 Resumen	78
PARTE III. DISEÑO Y DESARROLLO DEL MARCO.....	79
CAPÍTULO 4 RESUMEN DEL SISTEMA DE DIFUSIÓN SELECTIVA IMPLEMENTADO Y ARQUITECTURA FUNCIONAL	
DEFINIDA	79
4.1 Requisitos Funcionales de la Arquitectura	79
4.2 Conceptos de la Arquitectura Funcional Definida e Implementada.....	80

4.2.1	Terminología Específica en la Arquitectura Diseñada	80
4.2.2	Definición de Elementos Específicos en la Arquitectura Desarrollada	81
4.2.3	Espacios de Trabajo	84
4.2.3.1	Arquitectura en Espacio de Trabajo Simple.....	85
4.2.3.2	Arquitectura en Espacio de Trabajo Distribuido	86
4.3	<i>Arquitectura Funcional Definida para Aplicaciones de Difusión Selectiva</i>	88
4.3.1	Funcionamiento General del Sistema Sobre la Arquitectura Funcional	88
4.3.2	Arquitectura Funcional Auto-Proxy.....	91
4.4	<i>Alternativas de Arquitectura Funcional Valoradas</i>	93
4.4.1	Arquitectura Proxy-Caché	93
4.4.2	Arquitectura Distribuida para Provisión de Contenido: P2P	95
4.5	<i>Sincronización de la Hora en la Arquitectura Funcional</i>	95
4.6	<i>Aspectos de Seguridad en las Conexiones</i>	97
4.7	<i>Resumen</i>	98
CAPÍTULO 5	DESCRIPCIÓN DEL MARCO DISEÑADO Y DESARROLLADO	100
5.1	<i>Elección de la Tecnología Software</i>	100
5.1.1	Condiciones Iniciales y Elección de la Tecnología	101
5.1.2	Elección de las Aplicaciones Software para el Desarrollo del Marco.....	102
5.1.3	Servidor Web en los Equipos Principales: Apache, MySQL y PHP	105
5.2	<i>Resumen del Marco para Aplicaciones de Difusión Selectiva Implementado</i>	108
5.2.1	Marco Diseñado y Desarrollado para Establecer la Configuración.....	108
5.2.2	Marco Diseñado y Desarrollado para Proporcionar el Funcionamiento (Sincronización y Reproducción)	110
5.3	<i>Entorno de Desarrollo</i>	113
5.4	<i>Sistema Software Diseñado y Desarrollado</i>	116
5.4.1	Introducción a las Aplicaciones Presentes en los Equipos Principales (ESCC, ERC) y su Comunicación	116
5.4.1.1	Método de Comunicación entre Equipos Principales (ESCC, ERC)	116
5.4.1.2	Necesidades Software en los Equipos Principales (ESCC, ERC)	116
5.4.2	Configuración y Aplicaciones Comunes en los Equipos Principales (ESCC, ERC)	120
5.4.2.1	Configuración del Servidor Web en los Equipos Principales (ESCC, ERC).....	120
5.4.2.2	Configuraciones Relativas a la Interfaz de Red de lo Equipos Principales (ESCC, ERC) debidas a la Virtualización mediante la aplicación VMWARE.....	122
5.4.2.3	Aplicación Web para Proporcionar una Interfaz Gráfica de Usuario en los Equipos Principales (ESCC, ERC).....	125
5.4.3	Configuraciones y Aplicaciones en el ESCC para Proveer la Configuración y el Contenido a los ERCs	139
5.4.3.1	Sistema Operativo del ESCC	139
5.4.3.2	Secuencia de Configuración en el Inicio del ESCC.....	139
5.4.3.3	Configuraciones Específicas del ESCC.....	139
5.4.3.4	Lógica Funcional del ESCC en el Marco de Aplicaciones de Difusión Selectiva	143
5.4.4	Configuraciones y Aplicaciones en los ERCs para Proveer la Reproducción de los Archivos.....	158
5.4.4.1	Sistema Operativo de los ERCs	158
5.4.4.2	Secuencia de Configuración en el Inicio de los ERCs.....	159
5.4.4.3	Configuraciones Específicas de los ERCs	160
5.4.4.4	Lógica Funcional del ERC en el Marco de Aplicaciones de Difusión Selectiva Diseñado y Desarrollado	165
5.5	<i>Resumen</i>	179
CAPÍTULO 6	MODIFICACIÓN DEL MARCO PARA LA EMISIÓN/RECEPCIÓN DE STREAMING MEDIA	182
6.1	<i>Plugin VLC para Mozilla Firefox</i>	182
6.2	<i>Modificación del Marco con Streaming Media Multicast</i>	183
6.2.1	Modificación de la Arquitectura Funcional.....	184
6.2.2	Modificación del Marco para Proporcionar Configuración	185
6.2.3	Modificación del Marco para Proporcionar el Funcionamiento.....	186
6.2.3.1	Configuraciones y Aplicaciones en lado del ESCC para proveer Multicast.....	186
6.2.3.2	Configuraciones y Aplicaciones en lado de los ERCs para reproducir Multicast	189
6.3	<i>Otros Plugins para Mozilla</i>	190
6.4	<i>Resumen</i>	191

PARTE IV. RESULTADOS OBTENIDOS Y PRUEBAS REALIZADAS 193

CAPÍTULO 7	COMPARATIVA: ARQUITECTURA FUNCIONAL IMPLEMENTADA Y ARQUITECTURA FUNCIONAL DE REFERENCIA	193
7.1	<i>Administración de los Equipos</i>	193
7.2	<i>Aplicaciones de Difusión Selectiva</i>	193
7.2.1	Portal Web de Vídeo	194
7.2.2	Canal de TV Corporativo.....	195
7.2.3	Difusión Selectiva Interactiva	196
7.3	<i>Rol de los Equipos en las Arquitecturas Funcionales</i>	197
7.3.1	Equipos con Rol Servidor.....	197
7.3.2	Equipos con Rol Reproductor	198
7.4	<i>Métodos de Funcionamiento y Trabajo</i>	199
7.5	<i>Resumen</i>	200
CAPÍTULO 8	ENTORNO DE PRUEBAS.....	202
8.1	<i>Resumen del Entorno de Pruebas</i>	203
8.1.1	Equipamiento Disponible para Entorno de Pruebas	203
8.1.2	Aplicaciones Utilizadas en el Entorno de Pruebas.....	203
8.1.3	Espacios de Trabajo Evaluados en el Entorno de Pruebas.....	204
8.1.4	Aspectos a Considerar en el Entorno de Pruebas	205
8.2	<i>Esquema Lógico/Físico del Entorno de Pruebas</i>	205
8.2.1	Espacio de Trabajo Simple Virtualizado.....	206
8.2.2	Espacio de Trabajo Simple.....	207
8.2.3	Espacio de Trabajo Distribuido	209
8.2.4	Conexión Física mediante AP.....	211
8.2.5	Espacio de Trabajo Distribuido en Emisión Streaming Media Local	213
8.3	<i>Tipos de Pruebas Realizadas</i>	215
8.3.1	Espacio de Trabajo en modo NAP/NS (No Auto-Proxy y No Sincronización)	215
8.3.2	Espacio de Trabajo en modo NAP/CS (No Auto-Proxy y Con Sincronización): Peor Situación... ..	216
8.3.3	Espacio de Trabajo en modo AP/CS (Auto-Proxy y Con Sincronización)	217
8.3.4	Espacio de Trabajo en modo AP/NS (Auto-Proxy y No Sincronización): Modo Estable	217
8.4	<i>Resumen</i>	217
CAPÍTULO 9	RESULTADOS DE LAS PRUEBAS REALIZADAS.....	218
9.1	<i>Pruebas de Funcionamiento del Marco para Aplicación de Cartelería Digital y Técnica HTTP Streaming</i>	218
9.1.1	Espacio de Trabajo Simple Virtualizado: 0100X	223
9.1.2	Espacio de Trabajo Simple con Conexión Física. 0200X.....	229
9.1.3	Espacio de Trabajo Distribuido con 1 solo Equipo ERC. 0300X	233
9.1.4	Espacio de Trabajo Distribuido con 2 Equipos ERC. Configuración límite del BW entre sedes. 0400X	235
9.1.5	Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN Insuficiente en una sede: 0500X.....	238
9.1.6	Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN necesario en la sede del ESCC. 0600X.	241
9.2	<i>Prueba de Funcionamiento del Marco para Aplicación de Difusión Selectiva con Flujo Streaming Media Multicast</i>	243
9.2.1	Espacio de Trabajo Distribuido con 1 solo Equipo ERC.	243
9.2.2	Espacio de Trabajo Distribuido con 1 solo Equipo ERC. Live Streaming.....	245
9.3	<i>Pruebas Transversales</i>	245
9.3.1	Prueba Transversal: Secuencia de Inicio del ERC. T001	245
9.3.2	Prueba Transversal de Memoria Caché del Navegador. T002.....	252
9.3.3	Prueba Transversal Subjetiva de Calidad de Imagen en función de la resolución y del ancho de banda necesario en una sede. T003.	256
9.3.3.1	T003LQ: Calidad Baja de Imagen en los Archivos Reproducidos	260
9.3.3.2	T003MQ: Calidad Media de Imagen en los Archivos Reproducidos	263
9.3.3.3	T003HQ: Calidad Alta de Imagen en los Archivos Reproducidos	266
9.4	<i>Resumen</i>	269

PARTE V. CONCLUSIONES Y TRABAJOS FUTUROS	271
CAPÍTULO 10 CONCLUSIONES Y TRABAJOS FUTUROS.....	271
10.1 Conclusiones	271
10.2 Líneas de Trabajo Futuro	277
10.2.1 Nuevas Funcionalidades y Mejoras de la Aplicación de Cartelería Digital	277
10.2.2 Nuevos Servicios y Aplicaciones en el Marco. Mejoras Relacionadas con la Arquitectura	
Funcional	279
10.2.3 Implementación Difusión Selectiva Interactiva.....	281
10.2.4 Mejorar Aspectos Relacionados con la Seguridad.....	282
10.2.5 Instalación Automatizada.....	282
PARTE VI. APÉNDICES.....	283
A PRESUPUESTO	283
A.I. INTRODUCCIÓN	283
A.II. DESCOMPOSICIÓN EN FASES Y TAREAS.....	283
A.II.1. FASE 1: Documentación y Análisis del Estado de Arte	284
Tarea 1.1: Estudio de las Soluciones Existentes en el Mercado de la Difusión Selectiva	284
Tarea 1.2: Estudio en Detallado de la Solución de Referencia	284
Tarea 1.3: Selección de las Características Generales del Sistema a Implementar	285
Tarea 1.4: Estudio y Pruebas de las Tecnologías para Implementar un Sistema de Difusión Selectiva	
.....	285
A.II.2. FASE 2: Diseño del Marco para Aplicaciones de Difusión Selectiva	285
Tarea 2.1: Diseño de la Arquitectura Funcional para Aplicaciones de Difusión Selectiva	285
Tarea 2.2: Diseño del Marco para Aplicaciones de Difusión Selectiva.....	286
Tarea 2.3: Instalación y Configuración del Software para el Desarrollo del Marco para Aplicaciones	
de Difusión Selectiva	286
A.II.3. FASE 3: Desarrollo de una Interfaz Gráfica de Configuración del Servidor de Contenido.	
División en Capas	286
Tarea 3.1: Servidor. Elección de las Distintas Capas de Funcionamiento del Servidor	287
Tarea 3.2: Servidor. Capa 3: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo	
Servidor	287
Tarea 3.3: Servidor. Capa 2: Desarrollo de la Capa de Motor PHP del Equipo Servidor.....	287
Tarea 3.4: Servidor. Capa 1: Desarrollo de la Capa de la Base de Datos.....	288
Tarea 3.5: Servidor. Capa 0: Desarrollo de la Capa de Sistema	288
Tarea 3.6: Servidor. Integración y Pruebas de Comunicación Entre Capas.	288
A.II.4. FASE 4: Desarrollo del Interfaz de Configuración del Equipo Terminal Cliente. División	
en Capas del Equipo Reproductor Cliente	289
Tarea 4.1: Cliente. Elección de las Distintas Capas de Funcionamiento del Equipo Terminal Cliente	
.....	289
Tarea 4.2: Cliente. Capa 2: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo	
Cliente	289
Tarea 4.3: Cliente. Capa 1: Desarrollo de la Capa Base de datos	289
Tarea 4.4: Cliente. Capa 0: Desarrollo de la Capa de Sistema.....	290
Tarea 4.5: Cliente. Integración y Pruebas de Comunicación entre Capas.....	290
A.II.5. FASE 5: Diseño y Desarrollo de un Marco para la Provisión y Visualización de	
Contenido Multimedia.....	290
Tarea 5.1: Servidor y Cliente. Diseño de un Marco para Aplicaciones de Difusión Selectiva de	
Contenido Multimedia.....	290
Tarea 5.2: Servidor. Desarrollar un script en el Servidor que Atienda las Peticiones de Configuración	
de los Equipos Terminales Clientes	291
Tarea 5.3: Cliente. Desarrollar el Mecanismo por el cual el Equipo Terminal Cliente Presentará el	
Contenido Multimedia.....	291
Tarea 5.4: Servidor y Cliente. Desarrollar el Mecanismo que Permite al Cliente Detectar Cambios en	
la Configuración del Contenido Multimedia a Mostrar.....	291
Tarea 5.5: Servidor y Cliente. Verificar el Funcionamiento del Marco Diseñado y Desarrollado....	292
A.II.6. FASE 6: Análisis del Funcionamiento del Marco Diseñado y Desarrollado.....	292
Tarea 6.1: Instalación del Sistema Servidor y Sistema Cliente en Máquina Virtual	292
Tarea 6.2: Instalación de los Equipos de Red para Simular un Ambiente de Producción Real	292
Tarea 6.3: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Simple.....	293
Tarea 6.4: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Distribuido	
.....	293

A.II.7.	<i>FASE 7: Revisión y Mejora del Marco Desarrollo</i>	293
	Tarea 7.1: Evaluación de la Viabilidad de Posibles Mejoras del Marco Desarrollado	293
	Tarea 7.2: Desarrollo de las Mejoras Seleccionadas	294
	Tarea 7.3: Realización de Estudio Comparativo con Mejoras Introducidas en el Marco	294
A.II.8.	<i>FASE 8: Documentación y Redacción de la Memoria del Proyecto</i>	294
	Tarea 8.1: Documentación del Desarrollo del Proyecto y las Pruebas Realizadas.....	294
	Tarea 8.2: Redacción del Documento Final de Proyecto	294
A.III.	RESUMEN DEL PROYECTO	296
A.IV.	DIAGRAMA DE GANTT	298
A.V.	COSTES DEL PROYECTO	299
B	INSTALACIÓN Y CONFIGURACIÓN DEL ENTORNO DE PRUEBAS	301
B.I.	INSTALACIÓN DE SERVIDOR DE MÁQUINAS VIRTUALES. VMWARE SERVER	301
B.II.	CONFIGURACIÓN BW WAN EN ESPACIO DISTRIBUIDO CON VARIAS SEDES.....	303
B.III.	CONFIGURACIÓN DE LOS ROUTERS Y EL SWITCH	307
B.III.1.	Router A.....	308
B.III.2.	Router B.....	310
B.III.3.	Switch	311
B.IV.	CONFIGURACIÓN DE LA APLICACIÓN VLC COMO EMISOR MULTICAST	314
B.IV.1.	Elección de la fuente de contenido multimedia.	315
B.IV.2.	Destino de la Emisión	316
B.IV.3.	Establecer las Opciones	318
C	MANUAL DEL USUARIO ADMINISTRADOR	320
C.I.	MANUAL DEL USUARIO ADMINISTRADOR DEL ESCC	320
C.I.1.	<i>MENÚ UBICACIONES</i>	321
	UBICACIONES >> Seleccionar/Crear Ubicación	322
	UBICACIONES >> Configurar Ubicación >> Zonas a Esc	326
	UBICACIONES >> Configurar Ubicación >> Listas a Zonas.....	327
	UBICACIONES >> Configurar Ubicación >> Archivos a Listas.....	330
	UBICACIONES >> Equipos a Ubicaciones	333
C.I.2.	<i>MENÚ ARCHIVOS</i>	335
	ARCHIVOS >> Gestión de Archivos.....	336
	ARCHIVOS >> Búsqueda de Archivos	338
C.I.3.	<i>MENÚ ACTUALIZAR</i>	341
	ACTUALIZAR >> Contenidos en Ubicaciones.....	341
	ACTUALIZAR >> Actualizar Ubicaciones/Equipos y Sincronizar	343
C.I.4.	<i>MENÚ SISTEMA</i>	344
	SISTEMA >> Configuración Servidor.....	344
	SISTEMA >> Configuración Servidor >> Parámetros de Red	345
	SISTEMA >> Configuración Servidor >> Servidores DNS.....	345
	SISTEMA >> Configuración Servidor >> Referencias NTP.....	346
	SISTEMA >> Configuración Servidor >> Cambia Claves	346
	SISTEMA >> Estado Servidor.....	347
	SISTEMA >> Estado Servidor >> Estado Red	347
	SISTEMA >> Estado Servidor >> Estado DNS.....	349
	SISTEMA >> Estado Servidor >> Estado NTP	349
	SISTEMA >> Reiniciar	350
C.I.5.	<i>MENÚ DESCARGAS</i>	350
	DESCARGAS >> Enlaces Software.....	351
	DESCARGAS >> Archivos.....	352
C.II.	MANUAL DEL USUARIO ADMINISTRADOR DEL ERC	353
C.II.1.	<i>MENÚ CONFIGURACIÓN</i>	355
	CONFIGURACION >> Direccionamiento IP ESCC	356
	CONFIGURACION >> Ancho de Banda Sincro	356
C.II.2.	<i>MENÚ SISTEMA</i>	357
	SISTEMA >> Direccionamiento IP ERC.....	358
	SISTEMA >> Clave Admin/Clave Sistema.....	358
C.II.3.	<i>MENÚ ESTADO</i>	359
	ESTADO >> Estado Red Cliente	359

ESTADO >> PING Servidores	360
ESTADO >> Archivos Locales.....	362
D INSTALACIÓN DE MEDIA WIKI EN EL SISTEMA.....	364
E ARCHIVO DE POLÍTICAS URL PARA LAS APLICACIONES SWF	369
BIBLIOGRAFÍA Y REFERENCIAS	370
GLOSARIO	383

Índice de Ilustraciones

ILUSTRACIÓN 1.-. <i>UNICAST</i> . LA INFORMACIÓN SE TRASMITTE HACIA LOS NODOS DE FORMA INDEPENDIENTE EN CONEXIONES DIFERENTES	29
ILUSTRACIÓN 2.-. <i>MULTICAST</i> . LA INFORMACIÓN SE TRASMITTE HACIA LOS NODOS QUE PERTENECEN AL GRUPO DE MULTIDIFUSIÓN DE LA INFORMACIÓN ENVIADA.	29
ILUSTRACIÓN 3.-. <i>BROADCAST</i> . LA INFORMACIÓN SE ENVÍA DIRECTAMENTE A TODOS LOS ENLACES.....	30
ILUSTRACIÓN 4.-. ESQUEMA BÁSICO DE SEDES INTERCONECTADAS A TRAVÉS DE INTERNET MEDIANTE VPNs.....	31
ILUSTRACIÓN 5.-. ESQUEMA BÁSICO DE INTERCONEXIÓN DE SEDES A TRAVÉS DE LÍNEAS PRIVADAS DEDICADAS.	32
ILUSTRACIÓN 6.-. ESQUEMA BÁSICO DE UTILIZACIÓN DE ENLACES SATÉLITE. SE OBSERVA EN EL ESQUEMA LA POSIBILIDAD DE CONECTAR SEDES AISLADAS.....	33
ILUSTRACIÓN 7.-. DIFUSIÓN SELECTIVA PASIVA: LA INFORMACIÓN SOLAMENTE VA EN UN SENTIDO, EL CONTENIDO NO SE ADAPTA AL ENTORNO.	35
ILUSTRACIÓN 8.-. DIFUSIÓN SELECTIVA ACTIVA: LA INFORMACIÓN MOSTRADA POR EL DISPOSITIVO DE PRESENTACIÓN SE ADAPTA AL ENTORNO DETECTADO.	35
ILUSTRACIÓN 9.-. DIFUSIÓN SELECTIVA INTERACTIVA: EL USUARIO INTERCAMBIA CONTINUAMENTE INFORMACIÓN CON ALGÚN DISPOSITIVO SENSOR; ASÍ LA INFORMACIÓN MOSTRADA CAMBIA CONTINUAMENTE.	36
ILUSTRACIÓN 10.-. EVOLUCIÓN ESPERADA EN EL DESARROLLO DE UNA SOLUCIÓN PARA APLICACIONES DE DIFUSIÓN SELECTIVA.	56
ILUSTRACIÓN 11.-. SERVICIOS PROPORCIONADOS POR EL SISTEMA DE DIFUSIÓN SELECTIVA DE CISCO Y ELEMENTOS MÁS IMPORTANTES DENTRO DE LA ARQUITECTURA FUNCIONAL. FUENTE CISCO 2008-2010.	59
ILUSTRACIÓN 12.-. ELEMENTOS PRINCIPALES DEL SERVICIO DE CARTELERÍA DIGITAL DE CISCO. FUENTE: CISCO 2008-2010.	61
ILUSTRACIÓN 13.-. ELEMENTOS PRINCIPALES DEL SERVICIO DE PORTAL DE VIDEO DE CISCO. FUENTE: CISCO 2008-2010. ..	62
ILUSTRACIÓN 14.-. ELEMENTOS PRINCIPALES DEL SERVICIO DE SISTEMA DE TV CORPORATIVO DE CISCO. FUENTE: CISCO 2008-2010.	63
ILUSTRACIÓN 15.-. ARQUITECTURA PROPUESTA POR LA SOLUCIÓN DE CISCO (EJEMPLO PARA CISCO DIGITAL SIGNAGE). ESPACIO DE FUNCIONAMIENTO SIMPLE Y ESPACIO DE FUNCIONAMIENTO DISTRIBUIDO. FUENTE: CISCO 2008-2010.	64
ILUSTRACIÓN 16.-. ELEMENTOS QUE INTERVIENEN EN LA SOLUCIÓN DISTRIBUIDA PROPUESTA POR CISCO (CISCO DIGITAL SIGNAGE COMPONENTS). FUENTE CISCO, 2008-2010.	66
ILUSTRACIÓN 17.-. PORTAL DE VIDEO DE CISCO EN CONFIGURACIÓN DE ESPACIO DE FUNCIONAMIENTO SIMPLE EN LA SOLUCIÓN DE REFERENCIA. FUENTE: CISCO 2008-2010.	68
ILUSTRACIÓN 18.-. PORTAL DE VIDEO DE CISCO EN CONFIGURACIÓN DE ESPACIO DE FUNCIONAMIENTO DISTRIBUIDO EN LA SOLUCIÓN DE REFERENCIA. FUENTE: CISCO 2008-2010.	71
ILUSTRACIÓN 19.-. PORTAL DE VIDEO DE CISCO EN CONFIGURACIÓN DE ESPACIO DE FUNCIONAMIENTO DISTRIBUIDO EN LA SOLUCIÓN DE REFERENCIA CON CAPACIDAD PARA PROPORCIONAR VoD. FUENTE: CISCO 2008-2010.	73
ILUSTRACIÓN 20.-. PORTAL DE VIDEO DE CISCO EN CONFIGURACIÓN DE ESPACIO HÍBRIDO EN LA SOLUCIÓN DE REFERENCIA. FUENTE: CISCO 2008-2010.....	75
ILUSTRACIÓN 21.-. RESUMEN DE ARQUITECTURA DE DIFUSIÓN SELECTIVA DE FABRICANTE REFERENCIA. FUENTE: CISCO 2008-2010.	76
ILUSTRACIÓN 22.-. EJEMPLO DE USO DE PROTOCOLOS DE COMUNICACIÓN ENTRE LOS DIFERENTES ELEMENTOS DE LA ARQUITECTURA EN UN ESPACIO DE FUNCIONAMIENTO DISTRIBUIDO. FUENTE: CISCO 2010.....	77
ILUSTRACIÓN 23.-. ARQUITECTURA DEFINIDA	82
ILUSTRACIÓN 24.-. EJEMPLO DE ESPACIO SIMPLE EN LA ARQUITECTURA FUNCIONAL IMPLEMENTADA.	85
ILUSTRACIÓN 25.-. EJEMPLO DE ESPACIO SIMPLE DONDE APARECE UNA CONEXIÓN <i>WAN</i> PARA ESTABLECER LA CONFIGURACIÓN Y CARGA DEL CONTENIDO MULTIMEDIA.	86
ILUSTRACIÓN 26.-. EJEMPLO DE ESPACIO DISTRIBUIDO EN LA ARQUITECTURA FUNCIONAL IMPLEMENTADA.	87
ILUSTRACIÓN 27.-. EJEMPLO DE ESPACIO DISTRIBUIDO CON UN SÓLO ERC.....	87
ILUSTRACIÓN 28.-. RESUMEN DE FUNCIONAMIENTO DE LA ARQUITECTURA FUNCIONAL.....	88
ILUSTRACIÓN 29.-. ARQUITECTURA FUNCIONAL CON CARACTERÍSTICA AUTO-PROXY EN CADA UNO DE LOS ERCs.	92
ILUSTRACIÓN 30.-. ARQUITECTURA FUNCIONAL CON ELEMENTOS PROXY-CACHÉ.	94
ILUSTRACIÓN 31.-. ARQUITECTURA FUNCIONAL CON ELEMENTOS AUTO-PROXY Y DISTRIBUCIÓN DE CONTENIDO UTILIZANDO TÉCNICAS P2P.	95

ILUSTRACIÓN 32.- MARCO DISEÑADO Y DESARROLLADO PARA ESTABLECER LA CONFIGURACIÓN DE LOS EQUIPOS.	108
ILUSTRACIÓN 33.- MARCO DISEÑADO Y DESARROLLADO PARA EL FUNCIONAMIENTO DE LA APLICACIÓN DE DIFUSIÓN SELECTIVA.	110
ILUSTRACIÓN 34.- EQUIPO ANFITRIÓN CON VARIAS INSTANCIAS VIRTUALIZADAS DE LOS EQUIPOS ESCC Y ERC.	113
ILUSTRACIÓN 35.- SECUENCIA LÓGICA DE LOS ARCHIVOS PROGRAMADOS EN LENGUAJE <i>PHP</i> CORRESPONDIENTES A LA INTERFAZ DE USUARIO DEL ADMINISTRADOR DEL SISTEMA: ARCHIVOS <i>PHP</i> BASE.	127
ILUSTRACIÓN 36.- RESUMEN DEL FLUJO LÓGICO DEL BLOQUE QUE RESTRINGE EL ACCESO A LA CONFIGURACIÓN DEL SISTEMA.	130
ILUSTRACIÓN 37.- ESQUEMA GENERAL DEL FUNCIONAMIENTO MODULAR EN CAPAS, DESARROLLADO EN EL ESCC, DE LA ARQUITECTURA SOFTWARE PARA LA CONFIGURACIÓN DE LA APLICACIÓN DE DIFUSIÓN SELECTIVA.	144
ILUSTRACIÓN 38.- ESQUEMA GENÉRICO DE FUNCIONAMIENTO DE LAS APLICACIONES SWF EN EL ESCC.	148
ILUSTRACIÓN 39.- MENÚ DE CONFIGURACIÓN DE LAS APLICACIONES DE INICIO SOBRE UBUNTU 10. 10 BASADO EN GNOME.	161
ILUSTRACIÓN 40.- CONFIGURACIÓN DE MOZILLA FIREFOX AL INICIO DEL ERC.	162
ILUSTRACIÓN 41.- CONFIGURACIÓN DEL NAVEGADOR MOZILLA FIREFOX PARA QUE NO INICIE LA ÚLTIMA SESIÓN.	163
ILUSTRACIÓN 42.- ADD-ONS PARA MOZILLA FIREFOX QUE PERMITE MOSTRAR EL CONTENIDO DEL NAVEGADOR A PANTALLA COMPLETA.	164
ILUSTRACIÓN 43.- ESQUEMA GENERAL DE FUNCIONAMIENTO MODULAR DE LA ARQUITECTURA SOFTWARE DE CONFIGURACIÓN DESARROLLADA EN EL ERC.	167
ILUSTRACIÓN 44.- ESQUEMA DE FUNCIONAMIENTO DEL INTERFAZ GRÁFICO DE USUARIO.	168
ILUSTRACIÓN 45.- ESQUEMA LÓGICO DE FUNCIONAMIENTO DE LA PÁGINA DE LA PANTALLA DE INICIO PRESENTADA POR EL SPC EN CADA <i>ERC</i>	170
ILUSTRACIÓN 46.- PRESENTACIÓN GENERADA EN LA PANTALLA DEL <i>ERC</i> , EN AUSENCIA DE CONEXIÓN CON EL <i>ESCC</i> , AL INICIAR EL EQUIPO.	171
ILUSTRACIÓN 47.- RESUMEN DEL SCRIPT QUE GENERA LA PÁGINA DE PRESENTACIÓN DE CONTENIDO MULTIMEDIA (<i>repEscPHP.php</i>).	173
ILUSTRACIÓN 48.- RESUMEN DE LA APLICACIÓN REPRODUCTOR SWF. LA APLICACIÓN PERMITE REPRODUCIR EL CONTENIDO MULTIMEDIA DE ACUERDO A LA CONFIGURACIÓN ALMACENADA EN EL <i>ESCC</i>	175
ILUSTRACIÓN 49.- RESUMEN DE LA APLICACIÓN CONTROL SWF. EL CLIENTE SOCKET PERMITE RECIBIR LAS ÓRDENES GENERADAS DESDE EL <i>ESCC</i> LAS CUALES SE INTERPRETAN EN LA APLICACIÓN EN EL <i>ERC</i>	177
ILUSTRACIÓN 50.- SOLUCIÓN DE STREAMING PROPUESTA POR VIDEO LAN. ARQUITECTURA FUNCIONAL.	183
ILUSTRACIÓN 51.- ARQUITECTURA FUNCIONAL SIMPLIFICADA CON NUEVO EQUIPO SERVIDOR PARA PROPORCIONAR FLUJOS MULTICAST EN LA RED.	184
ILUSTRACIÓN 52.- RESUMEN DEL SCRIPT QUE GENERA LA PÁGINA DE PRESENTACIÓN DE CONTENIDO MULTIMEDIA MODIFICADA PARA INTRODUCIR Y HACER USO DEL PLUGIN VLC (<i>repEscPHP.php</i>).	187
ILUSTRACIÓN 53.- PLUGIN POR DEFECTO EN UBUNTU DESKTOP 10.10 EN MOZILLA FIREFOX.	189
ILUSTRACIÓN 54.- PLUGIN VLC 1.1.4 EN UBUNTU DESKTOP 10.10 PARA MOZILLA FIREFOX.	190
ILUSTRACIÓN 55.- PLUGIN PARA MOZILLA FIREFOX DEL REPRODUCTOR AMBULANT PARA SMIL.	191
ILUSTRACIÓN 56.- ESQUEMA LÓGICO DE CONFIGURACIÓN EN ESPACIO DE TRABAJO SIMPLE VIRTUALIZADO.	206
ILUSTRACIÓN 57.- ESQUEMA LÓGICO DEL ENTORNO DE PRUEBAS EN LA CONFIGURACIÓN DEL ESPACIO DE TRABAJO SIMPLE.	207
ILUSTRACIÓN 58.- ESQUEMA LÓGICO CON DETALLE DE IMPLEMENTACIÓN FÍSICA DEL ENTORNO DE PRUEBAS EN LA CONFIGURACIÓN DEL ESPACIO DE TRABAJO SIMPLE.	208
ILUSTRACIÓN 59.- CONEXIÓN FÍSICA DE EQUIPOS EN ESPACIO DE TRABAJO SIMPLE.	209
ILUSTRACIÓN 60.- ESQUEMA LÓGICO DE ESPACIO DE TRABAJO DISTRIBUIDO PARA 3 SEDES MÁS SEDE CENTRAL.	210
ILUSTRACIÓN 61.- ESQUEMA LÓGICO CON DETALLE DE IMPLEMENTACIÓN FÍSICA DEL ENTORNO DE PRUEBAS EN LA CONFIGURACIÓN DEL ESPACIO DE TRABAJO DISTRIBUIDO.	210
ILUSTRACIÓN 62.- ESQUEMA FÍSICO DEL ENTORNO DE PRUEBAS EN LA CONFIGURACIÓN DEL ESPACIO DE TRABAJO DISTRIBUIDO.	211
ILUSTRACIÓN 63.- ESQUEMA DE CONEXIÓN FÍSICA DE LA FONERA, EQUIPOS ANFITRIONES Y SWITCH.	212
ILUSTRACIÓN 64.- ESQUEMA LÓGICO CON DETALLE DE IMPLEMENTACIÓN FÍSICA DEL ENTORNO DE PRUEBAS EN LA CONFIGURACIÓN DEL ESPACIO DE TRABAJO DISTRIBUIDO PARA PRUEBAS DE MEJORAS INTRODUCIDAS.	214
ILUSTRACIÓN 65.- CONEXIÓN FÍSICA DEL ENTORNO DE PRUEBAS PARA TODOS LOS ESPACIOS DE TRABAJO DEFINIDOS.	215
ILUSTRACIÓN 66.- PRUEBA 01001. MARCO FORZADO A FUNCIONAMIENTO EN NO ARQUITECTURA AUTO-PROXY Y NO SINCRONIZACIÓN DE DATOS. ESPACIO DE TRABAJO SIMPLE SOBRE EQUIPOS VIRTUALIZADOS EN EL MISMO EQUIPO ANFITRIÓN. INTERFAZ DE RED VIRTUALIZADA EN LOS EQUIPOS.	224

ILUSTRACIÓN 67.-. DETALLE DE PRUEBA 01001. SE MUESTRAN LOS PAQUETES/SEG Y LOS K BYTES/SEG EN LA MISMA GRÁFICA	225
ILUSTRACIÓN 68.-. PRUEBAS 01002 Y 01003. ACTIVACIÓN DE LA SINCRONIZACIÓN DE CONTENIDO EN EL BLOQUE DE PRUEBAS 0100X.	226
ILUSTRACIÓN 69.-. DETALLE PRUEBA 01002	228
ILUSTRACIÓN 70.-. SEDE RÁFAGAS. EL FLUJO HTTP STREAMING DE LOS ARCHIVOS REPRODUCIDOS ES A RÁFAGAS.	231
ILUSTRACIÓN 71.-. SEDE CONSTANTE. EL FLUJO PSEUDO STREAMING NECESARIO DE LOS ARCHIVOS REPRODUCIDOS ES APROXIMADAMENTE CONSTANTE.	232
ILUSTRACIÓN 72.-. 03001. MODO DE FUNCIONAMIENTO EN MODO HTTP STREAMING.	234
ILUSTRACIÓN 73.-. RESULTADO DE LAS PRUEBAS 03002 (NAP/CS) Y 03003(AP/CS).	235
ILUSTRACIÓN 74.-. PRESENTACIÓN DE RESULTADOS DE LA PRUEBA 0400[123]. SE PRESENTA EL INTERCAMBIO DE TRÁFICO ENTRE EQUIPOS.	237
ILUSTRACIÓN 75.-. DETALLE DE TRÁFICO EN 04002 SEPARADO POR TRÁFICO RSYNC Y NO RSYNC (!RSYNC) PARA CADA ERC.	237
ILUSTRACIÓN 76.-. RESULTADO DE LAS PRUEBAS 0500[123]. ANCHOS DE BANDA UTILIZADOS A LO LARGO DE LA PRUEBA.	239
ILUSTRACIÓN 77.-. TRÁFICO DE LA PRUEBA 05002 DESGLOSADO EN FUNCIÓN DEL TIPO DE TRÁFICO ENTRE LA SEDE CENTRAL Y EL RESTO DE SEDES.	240
ILUSTRACIÓN 78.-. RESULTADO DE LAS PRUEBAS 0600[123]. ANCHOS DE BANDA UTILIZADOS A LO LARGO DE LA PRUEBA. ANCHO DE BANDA LIMITADO POR ENLACE SERIAL DE LOS ROUTERS	242
ILUSTRACIÓN 79.-. DETALLE DEL TRÁFICO CORRESPONDIENTE (SIN HTTP STREAMING EN ERC-2)	243
ILUSTRACIÓN 80.-. TRÁFICO MULTICAST CAPTURADO DE LA EMISIÓN DEL ARCHIVO 1 DE LA LISTA_1.	244
ILUSTRACIÓN 81.-. CAPTURAS DE ARCHIVOS EN FLUJO STREAMING MEDIA MULTICAST PARA LA LISTA_1 EMITIDOS DESDE VLC CON EL PERFIL POR DEFECTO: H.264 +AAC.	244
ILUSTRACIÓN 82.-. CAPTURA EN RED DE EMISIÓN EN DIRECTO MEDIANTE WEBCAM Y FLUJO MULTICAST.	245
ILUSTRACIÓN 83.-. CAPTURA DE TRÁFICO HTTP INTERCAMBIADO EN LA SECUENCIA DE ARRANQUE.	246
ILUSTRACIÓN 84.-. FLUJO EN LA SECUENCIA DE ARRANQUE DEL ERC.	247
ILUSTRACIÓN 85.-. CONFIGURACIÓN DE LA CACHE EN EL NAVEGADOR FIREFOX PARA LA REALIZACIÓN DE LAS PRUEBAS T002.	253
ILUSTRACIÓN 86.-. ESTADO DE LA CACHE DEL NAVEGADOR FIREFOX CON EL SISTEMA EN FUNCIONAMIENTO.	254
ILUSTRACIÓN 87.-. ARCHIVOS ALMACENADOS EN DISCO. ARQUITECTURA CACHE AUTO-PROXY IMPLEMENTADA CON LA CONFIGURACIÓN DEL NAVEGADOR FIREFOX.	255
ILUSTRACIÓN 88.-. CAPTURA OBTENIDA PARA PRUEBAS T002.	255
ILUSTRACIÓN 89.-. RESOLUCIÓN DE SPC Y ZONA CONFIGURADA PARA PRESENTAR LOS VÍDEOS EN LAS PRUEBAS.	257
ILUSTRACIÓN 90.-. CONFIGURACIÓN DE LA UBICACIÓN. PANTALLA ZONAS A ESCENARIOS.	259
ILUSTRACIÓN 91.-. DIAGRAMA DE GANTT DEL PROYECTO.	298
ILUSTRACIÓN 92.-. EDITOR DE RED VIRTUAL DE LA APLICACIÓN VMWARE SERVER. ASOCIACIÓN DE INTERFAZ VIRTUAL E INTERFAZ FÍSICA.	301
ILUSTRACIÓN 93.-. CONFIGURACIÓN PARA PROPORCIONAR A LAS MÁQUINAS VIRTUALES LA CAPACIDAD DE UTILIZAR LAS INTERFACES FÍSICAS.	302
ILUSTRACIÓN 94.-. CONFIGURACIÓN DEL HARDWARE VIRTUAL SOBRE LA ASOCIACIÓN DE LAS INTERFACES DEFINIDA.	302
ILUSTRACIÓN 95.-. DIFERENCIAS ENTRE CONFORMADO DE TRÁFICO Y APLICACIÓN DE POLÍTICA DE CALIDAD DE SERVICIO. FUENTE CISCO 2010.	303
ILUSTRACIÓN 96.-. ESQUEMA DEL ALGORITMO TOKEN BUCKET. FUENTE CISCO 2010.	304
ILUSTRACIÓN 97.-. ALGORITMO DE 2 CUBOS CON FICHAS (TOKEN BUCKET). FUENTE CISCO 2010.	305
ILUSTRACIÓN 98.-. PANTALLA DE SELECCIÓN DE LA FUENTE DEL CONTENIDO MULTIMEDIA EN LA APLICACIÓN VLC. ELECCIÓN DE ARCHIVO MULTIMEDIA LOCAL.	315
ILUSTRACIÓN 99.-. ASISTENTE DE NAVEGACIÓN PARA LA CONFIGURACIÓN DE EMISIÓN DE CONTENIDO MULTIMEDIA EN LA APLICACIÓN VLC.	316
ILUSTRACIÓN 100.-. SELECCIÓN DEL TIPO DE DESTINO A ESTABLECER.	317
ILUSTRACIÓN 101.-. CONFIGURACIÓN DEL TIPO DE DESTINO PARA LA EMISIÓN DEL CONTENIDO MULTIMEDIA MEDIANTE TRÁFICO STREAMING MEDIA.	318
ILUSTRACIÓN 102.-. OPCIONES FINALES DE LA EMISIÓN.	319
ILUSTRACIÓN 103.-. PANTALLA DE ACCESO DE CONFIGURACIÓN DEL SISTEMA.	321
ILUSTRACIÓN 104.-. MENÚ UBICACIONES EN EL ESCC.	322
ILUSTRACIÓN 105.-. PANTALLA DE PRESENTACIÓN DONDE SE SELECCIONA/CREA LA UBICACIÓN A CONFIGURAR.	322
ILUSTRACIÓN 106.-. PANTALLAS EMERGENTES EN LA CREACIÓN DE NUEVAS UBICACIONES EN EL SISTEMA.	324
ILUSTRACIÓN 107.-. PANTALLA EMERGENTE EN LA ELIMINACIÓN DE UBICACIONES EN EL SISTEMA.	324

ILUSTRACIÓN 108.-. PANTALLA EMERGENTE DE PREVISUALIZACIÓN EN FORMATO 4:3.....	325
ILUSTRACIÓN 109.-. SE HA PERDIDO LA CONEXIÓN CON EL SERVIDOR SOCKET. AL PASAR 30 SEGUNDOS SE VUELVE A INTENTAR OBTENIENDO EL ACCESO.	325
ILUSTRACIÓN 110.-. ORDEN ENVIADA DESDE EL ESCC. LOS ERCs ASOCIADOS A LA UBICACIÓN 'TEATRO' DEBEN PEDIR DE NUEVO LA CONFIGURACIÓN DEL UBICACIÓN.	326
ILUSTRACIÓN 111.-. PANTALLA DE DEFINICIÓN DE LAS ZONAS ASIGNADAS A CADA UNO DE LOS POSIBLES ESCENARIOS DE UNA UBICACIÓN.	326
ILUSTRACIÓN 112.-. PANTALLA DE ASOCIACIÓN DE LISTAS DE REPRODUCCIÓN A ZONAS ESPECÍFICAS DE ESCENARIOS EN CADA UBICACIÓN.	328
ILUSTRACIÓN 113.-. PANTALLA DE ASOCIACIÓN DE ARCHIVOS A LISTAS DE REPRODUCCIÓN ESPECÍFICAS DE CADA UBICACIÓN.	330
ILUSTRACIÓN 114.-. PANTALLA DE ASOCIACIÓN DE LOS ERCs A CADA UBICACIÓN.	333
ILUSTRACIÓN 115.-. PANTALLA DE GESTIÓN DE LOS ARCHIVOS EN EL ESCC.	336
ILUSTRACIÓN 116.-. DIÁLOGO DE CARGA DE ARCHIVOS AL SERVIDOR.....	338
ILUSTRACIÓN 117.-. BARRA DE PROGRESO DE CARGA DEL ARCHIVO EN EL SERVIDOR.....	338
ILUSTRACIÓN 118.-. PANTALLA DE ADMINISTRACIÓN PARA REALIZAR LA BÚSQUEDA DE ARCHIVOS DENTRO DE LAS LISTAS ASIGNADAS A LAS UBICACIONES.	339
ILUSTRACIÓN 119.-. RESULTADO DE BUSCAR UNA LISTA CON NOMBRE DUPLICADO EN EL SISTEMA.	340
ILUSTRACIÓN 120.-. PANTALLA DE ADMINISTRACIÓN PARA ENVIAR ORDENES DE ACTUALIZACIÓN A TODO EL CONJUNTO DE EQUIPOS SUSCRITOS A UNA UBICACIÓN.....	342
ILUSTRACIÓN 121.-. PANTALLA DE GESTIÓN DE ACTUALIZACIÓN PETICIONES Y ORDENES DE SINCRONISMO DE CONTENIDO.	343
ILUSTRACIÓN 122.-. INFORMACIÓN ESCRITA EN EL SOCKET CUANDO SE ACTUALIZA UNA UBICACIÓN (IZQUIERDA) Y CUANDO SE INDICA QUE ACTUALICE UN SÓLO ERC (DERECHA).....	344
ILUSTRACIÓN 123.-. PANTALLA DE EDICIÓN DE LOS PARÁMETROS DE RED DEL ESCC.	345
ILUSTRACIÓN 124.-. EDICIÓN DE LOS SERVIDORES DNS EN EL ESCC.	346
ILUSTRACIÓN 125.-. EDICIÓN DE LAS REFERENCIAS NTP EN EL ESCC.	346
ILUSTRACIÓN 126.-. PANTALLA DE EDICIÓN DE CAMBIO DE CONTRASEÑA PARA LOS USUARIOS 'ADMIN' Y 'SISTEMA'.	347
ILUSTRACIÓN 127.-. PANTALLA QUE MUESTRA EL ESTADO DE LOS PARÁMETROS DE RED	348
ILUSTRACIÓN 128.-. RESULTADO DE VERIFICAR LA CONFIGURACIÓN DE LOS SERVIDORES DNS Y LA CONECTIVIDAD A ESTOS MEDIANTE PING.	349
ILUSTRACIÓN 129.-. ESTADO DE LOS TRES SERVIDORES NTP CONFIGURADOS.	350
ILUSTRACIÓN 130.-. HIPERENLACES A PROGRAMAS ÚTILES PARA ADMINISTRAR EL SISTEMA.....	351
ILUSTRACIÓN 131.-. LISTADO DE ARCHIVOS MULTIMEDIA ALMACENADOS EN EL ESCC.....	352
ILUSTRACIÓN 132.-. INTERFAZ DE ACCESO AL ERC.	354
ILUSTRACIÓN 133.-. CONFIGURACIÓN DE LA DIRECCIÓN IPV4 DEL ESCC DEL SISTEMA AL QUE PERTENECE EL ERC.....	356
ILUSTRACIÓN 134.-. CONFIGURACIÓN DEL ANCHO DE BANDA UTILIZADO PARA SINCRONIZAR EL CONTENIDO MULTIMEDIA DESDE EL ESCC AL ERC.	357
ILUSTRACIÓN 135.-. MENÚ SISTEMA.....	357
ILUSTRACIÓN 136.-. PÁGINA DE CONFIGURACIÓN DEL DIRECCIONAMIENTO IPV4 DEL ERC.	358
ILUSTRACIÓN 137.-. ACCESO AL CAMBIO DE CLAVES DE LOS USUARIOS DEFINIDOS EN EL SISTEMA.	359
ILUSTRACIÓN 138.-. MENÚ ESTADO DEL ERC.....	359
ILUSTRACIÓN 139.-. ESTADO DE LOS PARÁMETROS DE RED DEL ERC.	360
ILUSTRACIÓN 140.-. RESPUESTA DEL ESCC ANTE UN PING DESDE EL ERC CUANDO EXISTE CONECTIVIDAD.	361
ILUSTRACIÓN 141.-. RESPUESTA DEL ESCC ANTE UN PING DESDE EL ERC CUANDO NO EXISTE CONECTIVIDAD.	361
ILUSTRACIÓN 142.-. ACCESO A LA BÚSQUEDA DE ARCHIVOS MULTIMEDIA ALMACENADOS EN EL ERC.	362
ILUSTRACIÓN 143.-. LISTADO DE IMÁGENES DESCARGADAS DE FORMA LOCAL EN UN EQUIPO ERC.....	363
ILUSTRACIÓN 144.-. CREACIÓN DE LA BASE DE DATOS PAR APLICACIÓN MEDIAWIKI.	365
ILUSTRACIÓN 145.-. COMPROBACIÓN DE CREACIÓN DE LA BASE DE DATOS MEDIANTE PHPADMIN.	365
ILUSTRACIÓN 146.-. PANTALLA DE INICIO DE LA INSTALACIÓN DE MEDIAWIKI.	366
ILUSTRACIÓN 147.-. RESULTADO DE LA INSTALACIÓN MEDIAWIKI.....	367
ILUSTRACIÓN 148.-. PÁGINA PRINCIPAL DE LA APLICACIÓN MEDIA WIKI.	367

Índice de Tablas

TABLA 1.-. RESUMEN DE PROTOCOLOS DE CODIFICACIÓN (CÓDEC).....	41
TABLA 2.-. CONTENEDORES MULTIMEDIA.	42
TABLA 3.-. TIPO DE CODIFICACIÓN SOPORTADA POR LOS FORMATOS DE ARCHIVOS QUE REPRODUCE ADOBE FLASH PLAYER. .	44
TABLA 4.-. LISTADO DE EMPRESAS DE CARTELERÍA DIGITAL EN OPEN DIRECTORY PROJECT Y OTRAS SOLUCIONES DISPONIBLES.	49
TABLA 5.-. CARACTERÍSTICAS GENERALES DE LAS APLICACIONES DE DIFUSIÓN SELECTIVA ATENDIENDO AL PROPÓSITO ESPECÍFICO O AL CARÁCTER GENERAL	52
TABLA 6.-. CARACTERÍSTICAS GENERALES DE LAS ARQUITECTURAS ATENDIENDO A SI ESTABLECEN UN SISTEMA CLIENTE-SERVIDOR BASADO EN PETICIONES O UNA ARQUITECTURA DISTRIBUIDA.	52
TABLA 7.-. CARACTERÍSTICAS GENERALES DE LAS ARQUITECTURAS ATENDIENDO A SI ESTABLECEN UN AUTO-PROXY O BASADO EN PETICIONES DE CONTENIDO BAJO DEMANDA.....	54
TABLA 8.-. CARACTERÍSTICAS GENERALES DE LAS ARQUITECTURAS ATENDIENDO AL TIPO DE <i>HARDWARE</i> UTILIZADO PARA LOS EQUIPOS DEL SISTEMA FINAL.	56
TABLA 9.-. CONFIGURACIONES RECOMENDADAS SEGÚN CODIFICADORES DISPONIBLES EN LA SOLUCIÓN DE REFERENCIA. FUENTE CISCO 2008-2010.	60
TABLA 10.-. RESUMEN DE CARACTERÍSTICAS <i>HARDWARE</i> DE LOS DIFERENTES TIPOS DE CODIFICADORES DE VIDEO PROPORCIONADOS POR EL FABRICANTE DE REFERENCIA. FUENTE: CISCO 2008-2010.	61
TABLA 11.-. DEFINICIÓN DE ELEMENTOS QUE INTERVIENEN EN LA ARQUITECTURA.	83
TABLA 12.-. RESUMEN DE ARCHIVOS PHP BASE EN EL ESCC Y RELACIÓN DE ACCESO DESDE CADA USUARIO DEFINIDO EN LA APLICACIÓN DE DIFUSIÓN SELECTIVA DESARROLLADA.	135
TABLA 13.-. RESUMEN DE ARCHIVOS PHP BASE EN EL ERC Y RELACIÓN DE ACCESO DESDE CADA USUARIO DEFINIDO EN LA APLICACIÓN DE DIFUSIÓN SELECTIVA DESARROLLADA.	136
TABLA 14.-. RESUMEN DE INTERACCIÓN ENTRE CAPAS DE LOS ARCHIVOS Y APLICACIONES DESARROLLADAS EN EL PROYECTO PARA LA CONFIGURACIÓN DEL ESCC.	147
TABLA 15.-. RESUMEN DE SCRIPTS PHP QUE FORMAN EL MOTOR PHP	150
TABLA 16.-. RESUMEN DE ARCHIVOS QUE CONFORMAN LA INTERFAZ GRÁFICA DE USUARIO EN EL ERC.	169
TABLA 17.-. EQUIPAMIENTO UTILIZADO EN EL ENTORNO PRUEBAS.....	203
TABLA 18.-. APLICACIONES UTILIZADAS EN EL ENTORNO DE PRUEBAS.	204
TABLA 19.-. NECESIDADES MÍNIMAS DE <i>ANCHO DE BANDA</i> EN LISTAS DE REPRODUCCIÓN UTILIZADAS EN LAS PRUEBAS SEGÚN APLICACIÓN MEDIAINFO.....	218
TABLA 20.-. CARACTERÍSTICAS DE LOS ARCHIVOS UTILIZADOS EN LAS LISTAS DE REPRODUCCIÓN CONFIGURADAS EN LAS PRUEBAS. DATOS OBTENIDOS POR LA APLICACIÓN MEDIAINFO.	220
TABLA 21.-. NOTACIÓN DE PRUEBAS PARA LA EVALUACIÓN DE LA APLICACIÓN DE CARTELERÍA DIGITAL SOBRE EL MARCO Y PARA LA EVALUACIÓN DE CARACTERÍSTICA AUTO-PROXY SOBRE LA ARQUITECTURA FUNCIONAL.	220
TABLA 22.-. RESUMEN DE LAS PRINCIPALES PRUEBAS REALIZADAS PARA LA EVALUACIÓN DE LA APLICACIÓN DE CARTELERÍA DIGITAL DESARROLLADA SOBRE EL MARCO.	222
TABLA 23.-. RESUMEN DE LAS PRUEBAS REALIZADAS PARA LA EVALUACIÓN DE LA MEJORA STREAMING MEDIA MULTICAST SOBRE EL MARCO.	222
TABLA 24.-. RESUMEN DE LAS PRUEBAS TRANSVERSALES REALIZADAS.	223
TABLA 25.-. CÁLCULO DEL ÁREA DEL ARCHIVO 1 EN VARIOS INTERVALOS (RÁFAGA) DE LA PRUEBA 01001.....	224
TABLA 26.-. CÁLCULO DEL ÁREA DEL ARCHIVO 2 EN VARIOS INTERVALOS (RÁFAGA) DE LA PRUEBA 01001.....	225
TABLA 27.-. ANCHO DE BANDA DISPONIBLE Y NECESARIO EN CADA SEDE EN GRUPO DE PRUEBAS 0200X.	230
TABLA 28.-. CÁLCULO DEL ÁREA DEL ARCHIVO 1 (LISTA_1) EN VARIOS INTERVALOS (RÁFAGA) DE LA PRUEBA 02001.	231
TABLA 29.-. ANCHO DE BANDA DISPONIBLE Y NECESARIO EN CADA SEDE EN GRUPO DE PRUEBAS 0300X PARA CADA SEDE.	234
TABLA 30.-. ANCHO DE BANDA DISPONIBLE Y NECESARIO EN CADA SEDE EN GRUPO DE PRUEBAS 0500X PARA CADA SEDE.	236
TABLA 31.-. ANCHO DE BANDA DISPONIBLE Y NECESARIO EN CADA SEDE EN GRUPO DE PRUEBAS 0500X PARA CADA SEDE.	238
TABLA 32.-. ANCHO DE BANDA DISPONIBLE Y NECESARIO EN CADA SEDE EN GRUPO DE PRUEBAS 0600X PARA CADA SEDE.	241
TABLA 33.-. RESUMEN DE ARCHIVOS EVALUADOS EN LAS PRUEBAS SUBJETIVAS.	256
TABLA 34.-. COEFICIENTES Y RATIOS SEGÚN LA RESOLUCIÓN DE LOS VÍDEOS Y LA PANTALLA.....	257
TABLA 35.-. ELECCIÓN PORCENTAJE VERTICAL.	259
TABLA 36.-. RESULTADOS SUBJETIVOS OBTENIDOS EN VISUALIZACIÓN DE VÍDEOS CON DIFERENTES CALIDADES (RESOLUCIONES).	260

TABLA 37.-. RESUMEN DEL PROYECTO.	297
TABLA 38.-. REFERENCIA TOMADA PARA EL CÁLCULO DE LOS COSTES DE PERSONAL.	299

Resumen

En el mercado de las telecomunicaciones uno de los principales objetivos es el de proveer nuevos servicios, siendo, la *Ingeniería de Servicios*, uno de los principales retos en la investigación y desarrollo de aplicaciones. Normalmente, inicialmente se plantea un nuevo servicio de acuerdo a una necesidad detectada y, a continuación, se invierten horas de trabajo en desarrollar nuevas tecnologías o se usan otras existentes para desarrollar las aplicaciones que puedan proporcionar el servicio.

En este proyecto se ha afrontado el desarrollo de un sistema de difusión de contenido multimedia de acuerdo a un parámetro diferenciador que posibilite establecer reproducciones distintas en diferentes emplazamientos físicos. El sistema permite, a un administrador, gestionar la programación de forma remota y centralizada a través de una interfaz gráfica. Además, se ha diseñado la aplicación para que aborde el uso del *ancho de banda* de forma eficiente. El desarrollo de un sistema de estas características se espera que tenga una gran aceptación, ya que se ha detectado la necesidad de, por parte de las organizaciones, establecer sistemas de emisión de contenido digital multimedia, mediante pantallas y otros mecanismos, hacia una audiencia que se encuentre frente a estos medios de difusión. Se ha observado también que, las organizaciones de un cierto tamaño, suelen estar distribuidas geográficamente en diferentes sedes y que, normalmente, tienen un sistema de comunicación interno mediante redes privadas. Estas redes privadas son utilizadas para proporcionar otros servicios; no obstante, pueden ser aprovechadas para incluir otros nuevos, como el que se abarca en este proyecto, siempre que la utilización del *ancho de banda* sea adecuada.

Existen muchas propuestas comerciales que permiten difundir contenido multimedia de forma remota y que tienen la capacidad de programar estos contenidos en función de ciertos parámetros diferenciadores. Este tipo de propagación de contenido, de acuerdo a ciertos parámetros, se considera *Difusión Selectiva*; sin embargo, se ha detectado que un alto porcentaje de las soluciones comerciales analizadas, están orientadas a la difusión de contenido multimedia a modo de *Cartelería Digital* y que son pocas las empresas que abarcan el problema de forma general incluyendo la posibilidad de realizar emisiones en directo.

Por las razones anteriormente expuestas, el objetivo de este proyecto fin de carrera es desarrollar un sistema de *Difusión Selectiva* que abarque el problema de forma general. De este modo, se ha desarrollado un *Marco* como sustento para aplicaciones de *Difusión Selectiva* y no sólo como aplicación de *Cartelería Digital*. Así, se muestra el diseño y desarrollo de un *Marco* en el que, aglutinando diferentes tecnologías y desarrollando varias aplicaciones, se construye un servicio que permite difundir contenido multimedia de acuerdo a unos parámetros o unas características de la audiencia objetivo. En el contexto del *Marco* se desarrolla, también, una aplicación de *Cartelería Digital* que permite reproducir videos e imágenes de acuerdo a una programación. Esta aplicación de *Cartelería Digital* nos permitirá evaluar el *Marco*. En este trabajo se muestra, también, cómo pueden incluirse nuevas aplicaciones que permiten incluso la difusión de contenido en directo sin la modificación excesiva del *Marco* propuesto.

El *Marco* se ha desarrollado principalmente mediante el uso de tecnologías web, incluso para la presentación del contenido multimedia. Esto permite desarrollar e introducir un reproductor de contenido multimedia en un navegador y, por tanto, añadir la capacidad de reproducir *Streaming Media*, entre otras posibilidades, resulta inmediato si proporcionamos de esta capacidad al reproductor utilizado. Por tanto, el mecanismo adoptado para recuperar el contenido multimedia en el *Marco* se puede considerar muy flexible. En el desarrollo del

Marco se ha optado por utilizar *HTTP Streaming*, para proveer la reproducción de los videos e imágenes en una aplicación de *Cartelería Digital* y se ha optado, para salvaguardar el *ancho de banda*, por otras técnicas: sincronización de contenido multimedia en los equipos terminales y *arquitectura funcional* con característica *Auto-Proxy* definida en el proyecto.

Para la realización de pruebas, lo más acorde posible a una situación real, se diseñó y configuró un entorno de pruebas con *hardware* real y equipos virtualizados, que permitía simular diferentes contextos sin realizar apenas modificaciones. Esto permitió testear el *Marco* en muchos aspectos, además de confirmar el correcto funcionamiento.

Se han instalado y configurado diversas aplicaciones para así obtener una integración completa entre las aplicaciones desarrolladas y otras utilizadas. Esto ha creado la necesidad comprender algunos aspectos de funcionamiento de sistemas operativos basados en *Linux*. Además, se ha hecho un uso importante de aplicaciones de virtualización de equipos, realizando las configuraciones oportunas sobre los equipos anfitriones, tanto para desarrollar el *Marco*, como para la realización de las pruebas.

Se han adquirido importantes conocimientos acerca de sistemas para servidores web, tecnologías web específicas y propietarias, configuración de equipos de red, sistemas para la virtualización de equipos y protocolos y mecanismos para el transporte de contenido multimedia.

Con el ánimo de proporcionar al lector una visión en funcionamiento, del *Marco* diseñado y desarrollado, se ha creado un canal de acceso público en la plataforma *youtube* (<http://www.youtube.com/dtimonPFC>) con una gran variedad de videos que pueden ser visualizados.

PARTE I. Introducción

Capítulo 1 Introducción

Actualmente se ha detectado una necesidad de comunicación por parte de las organizaciones, ya sean empresas u organismos públicos, para acercarse a los ciudadanos de una forma más directa y más impactante con fines comerciales, de comunicación o informativos. Desde hace unos años se ha observado un cambio significativo en las organizaciones a la hora de presentar los mensajes al público, incluyendo clientes, empleados, socios y estudiantes. Algunas organizaciones están dejando de presentar sus mensajes de forma impresa y han migrado a una presentación de contenidos dinámicos, más flexibles, personalizables y en formato digital. El *Marketing* y la *Publicidad* se están dirigiendo al medio electrónico, ya no sólo por las ventajas e impacto que podría suponer, si no por las costumbres de los ciudadanos en la nueva era digital que nos ha tocado vivir. Las organizaciones financieras, por ejemplo, como bancos y cajas de ahorros, con sucursales de venta al público, aprovechan esta condición para mostrar productos y servicios en sus sucursales, creando, se supone, experiencias más ricas e interactivas para sus clientes. Otras industrias, incluyendo a los *gobiernos*, las *universidades* (y educación en general), *atención a la salud* (hospitales) y el *transporte*; también ponen en práctica la emisión de contenido multimedia como un instrumento para realzar y mejorar los mensajes a sus clientes o usuarios finales. De esta forma se están produciendo nuevos empleos y demandas sobre la red de infraestructuras de la propia organización.

El presente documento muestra el trabajo realizado para proporcionar un nuevo servicio mediante la configuración y desarrollo de diversas aplicaciones sobre las infraestructuras de red de las organizaciones. El *proyecto fin de carrera* tiene como objetivo principal el diseño y desarrollo de un *Marco* funcional que aprovisiona, a un sistema de comunicaciones, de los elementos principales para ofrecer un servicio de difusión de contenido multimedia, de acuerdo a algún parámetro o característica de la audiencia objetivo.

El desarrollo de este *Marco* permitirá, a una organización en general cuyos emplazamientos físicos de las distintas sedes estén distribuidos geográficamente e interconectados por diferentes tipos de redes de comunicaciones, cubrir estas nuevas necesidades.

Para el diseño y desarrollo del *Marco* para aplicaciones de *Difusión Selectiva* se ha indagado sobre las necesidades y se ha analizado el problema desde diferentes perspectivas. Se ha valorado qué solución podría ser la más recomendable para los recursos de los que se dispone en una hipotética organización y al mercado al que puede ir enfocada la aplicación sin encontrar una solución ideal. Así, dentro de este trabajo, entran en juego otras habilidades a desarrollar, entre ellas valorar las distintas soluciones encontradas, estudiar el funcionamiento de éstas y ver cuáles son las características y opciones a implementar dentro del proyecto. En el desarrollo del *Marco* influirán aspectos tales como, el método utilizado para distribuir el contenido multimedia o el mecanismo proporcionado al administrador del sistema para programar la difusión de los videos e imágenes en los distintos emplazamientos físicos.

Debido a la carencia de un léxico común a todos los proveedores de soluciones de *Difusión Selectiva*, en el presente trabajo, también, se ha llegado a definir algunos conceptos que se presentan difusos en la bibliografía consultada, como son los tipos de *Difusión Selectiva* o la

definición de cada uno de los nombres de los equipos involucrados en el sistema. Estas definiciones han sido muy útiles, tanto para explicar la *arquitectura funcional*, como el *Marco*.

Además, un aspecto fundamental es hacer un análisis a priori de las diferentes tecnologías que se ofrecen para implementar un sistema de estas características, de las limitaciones que puede llegar a imponer las diferentes elecciones y contrastarlas con las ventajas que supone un tipo de tecnología con respecto a las descartadas.

Una vez establecida y escogida una determinada tecnología, se intenta dividir el proceso de implementación en diferentes capas. Se intenta hacer cada capa lo más independiente posible de la anterior, algo similar a lo que ocurre en la implementación de una pila de protocolo de comunicaciones, pero todo ello a nivel de aplicación. Así, el hecho de cambiar la interfaz de usuario no implicaría cambiar las bases de datos que sustentan la configuración de los equipos y un cambio en las bases de datos no implicaría un cambio radical en la interfaz de usuario.

Una vez finalizada la implementación del sistema se ha probado y testeado con varios objetivos: encontrar anomalías en la arquitectura desarrollada e identificar nuevas mejoras en las funcionalidades o en la distribución de contenido, cerrando, como en cualquier aplicación desarrollada, el círculo para la mejora del servicio.

Por último, se han identificado los problemas principales que surgen en conceptos de seguridad, para así, en futuros desarrollos realizar una retroalimentación salvando dichas barreras.

1.1 *Objetivos del Proyecto*

Se puede realizar un resumen de los objetivos más importantes del proyecto en una serie de puntos:

- Estudio de las soluciones de *Difusión Selectiva* encontradas en el mercado.
- Estudio del estado del arte actual en aspectos relacionados con la *Difusión Selectiva*.
- Estudio de tipos de soluciones encontradas. Aspectos relevantes en el funcionamiento de una arquitectura de *Difusión Selectiva*.
- Análisis de alternativas de implementación de una arquitectura funcional. Ventajas e Inconvenientes de las distintas opciones.
- Identificar aspectos relacionados con la seguridad del sistema desarrollado.
- Diseño de una arquitectura funcional de *Difusión Selectiva*.
- Estudio de las alternativas tecnológicas para el desarrollo de la *arquitectura funcional* por medio de un *Marco*.
- Diseño de un *Marco* escalable para aplicaciones de *Difusión Selectiva* de *Contenido Multimedia*.
- Adquisición de conocimientos en la instalación, integración y uso, de las aplicaciones necesarias y complementarias para el funcionamiento de la arquitectura.
- Adquisición de conocimientos de programación relacionados con las tecnologías utilizadas.
- Desarrollo de un *Marco* para aplicaciones de *Difusión Selectiva* de *Contenido Multimedia*. Desarrollo de una aplicación en un servidor que controle la configuración de la *arquitectura funcional* para aplicaciones de *Difusión Selectiva*.

de contenido multimedia. Desarrollo de una aplicación con capacidad para mostrar el contenido multimedia de forma remota a través de medios de presentación.

- Adquisición de conocimiento para plantear y configurar un escenario de pruebas apropiado.
- Realización de pruebas basadas en un entorno lo más parecido posible a situaciones reales de funcionamiento.
- Extracción de conclusiones acerca de la arquitectura implementada, así como la evaluación del sistema en funcionamiento sobre el *Marco*.

1.2 Estructura de la Memoria

La estructura de este documento se ha dividido en varias partes:

Parte I: Introducción

Capítulo 1. Introducción: Este capítulo presenta una introducción general al proyecto, se enumeran los objetivos de éste y se presenta la estructura en la que se ha organizado la memoria.

Parte II: Estado del Arte. Soluciones Comerciales

Se realizan definiciones y consideraciones sobre el concepto de *Difusión Selectiva* y se realiza un estudio de las soluciones comerciales existentes. Esta parte se divide en diversos capítulos:

Capítulo 2. Conceptos y Consideraciones de Difusión Selectiva. En este capítulo se establecen las definiciones y los aspectos más generales del proyecto relacionados con la *Difusión Selectiva*. Se realiza una breve introducción de los elementos que pueden influir en un sistema de este tipo y se presentan los aspectos más importantes y de consideración que se deben tener presentes a la hora de diseñar el *Marco*.

Capítulo 3. Soluciones para Proporcionar Servicios de Difusión Selectiva. Se recoge, con la definición de varios tipos de *arquitecturas funcionales* y métodos para alcanzar aplicaciones de *Difusión Selectiva*, un resumen de todas las soluciones comerciales encontradas en el mercado. Se observa que, la mayoría de los fabricantes, se vuelcan en cierta aplicación más comercial denominada *Cartelería Digital*. Por esta razón, nos centramos en el estudio detallado de una solución a la que se considera de referencia por abarcar el problema de forma general. Se presenta, en detalle, la solución considerada como de referencia y se muestran ciertos modos de funcionamiento que afectan al diseño de una determinada solución utilizando el sistema de referencia.

Parte III: Diseño y Desarrollo del Marco

Se describe el trabajo realizado dividiendo esta parte en varios capítulos directamente relacionados, debido a que el diseño del *Marco* depende fuertemente de la *arquitectura funcional* ideada. Se muestra el desarrollo atendiendo a las tecnologías seleccionadas. Se divide en varios capítulos:

Capítulo 4. Resumen del Sistemas de Difusión Selectiva Implementado y Arquitectura Funcional Definida. En este capítulo se establecen algunos requisitos funcionales de la arquitectura y se realizan las definiciones específicas de ciertos conceptos que serán utilizados a lo largo de la memoria para explicar el trabajo realizado y los resultados

obtenidos. Se presenta la *arquitectura funcional implementada* y se muestran algunas alternativas a ésta.

Capítulo 5. Descripción del Marco Diseñado y Desarrollado. Según la *arquitectura funcional* definida, la cual describe el comportamiento del sistema desarrollado, se diseña el *Marco* para aplicaciones de *Difusión Selectiva*. Se enumeran ciertos prerequisites y se muestran algunos motivos de la elección de la tecnología para desarrollar el *Marco*. También, en este capítulo, se muestran las configuraciones necesarias a partir de la estrategia del *Marco* diseñado y se resumen las aplicaciones desarrolladas de modo conceptual. El capítulo se divide en varios subapartados debido a que son varios tipos de equipos los que intervienen en la *arquitectura funcional* y tienen importantes diferencias y funciones.

Capítulo 6. Mejoras en el Marco: Propuesta Emisión/Recepción Streaming Media. En el capítulo se muestran cómo se pueden incluir nuevas funcionalidades o aplicaciones al *Marco* desarrollado realizando pequeñas modificaciones. Este capítulo evidencia el buen diseño del *Marco*, el cual se presenta como muy flexible y permite incluir nuevas aplicaciones de forma relativamente sencilla.

Parte IV: Resultados Obtenidos y Pruebas Realizadas

Se ha considerado que las pruebas sobre el *Marco* deben ajustarse a situaciones de trabajo reales, por lo que se le ha dado gran importancia a las pruebas realizadas con el objetivo de testear de forma rigurosa el *Marco*. Son varios los capítulos en los que se ha dividido esta parte:

Capítulo 7. Comparativa: Arquitectura Funcional Implementada y Arquitectura Funcional de Referencia. Se establece una comparativa entre la *arquitectura funcional implementada* y la arquitectura de referencia en los aspectos comunes y se remarcan algunas características y diferencias. Esta comparativa se establece para verificar el buen diseño de la *arquitectura funcional* y el *Marco* en general.

Capítulo 8. Entorno de Pruebas. Se define un *entorno de pruebas* para cada una de los *espacios de trabajo* que se pretenden evaluar. En el capítulo se muestra cómo se conectan una serie de equipos que permite generar un entorno de pruebas apropiado y evaluar el sistema en las situaciones de interés sin la necesidad de realizar cambios importantes en la configuración de dichos equipos. Se hace uso de la virtualización, además de la conexión y configuración apropiada de varios equipos electrónicos, de modo que se establece un entorno de pruebas lo suficientemente realista para testear de modo riguroso el *Marco*. En el capítulo se describen, también, los tipos de pruebas realizadas y se muestra cómo se ha forzado al *Marco* para que trabaje de forma similar a como lo hacen otras *arquitecturas funcionales* en las que no se intenta preservar el *ancho de banda*. De este modo, se compara el comportamiento de la *arquitectura funcional* del *Marco* desarrollado con el de otras menos sofisticadas.

Capítulo 9. Resultados de la Pruebas Realizadas. En este capítulo se detallan los resultados de evaluar el *Marco* en el entorno de pruebas definido en el Capítulo 8. Se muestran resultados a considerar, tanto cuantitativos como cualitativos, con el *Marco* funcionando en diversos modos. En los resultados de las pruebas realizadas, también se muestra la confirmación del funcionamiento, en el *Marco* desarrollado, de las mejoras propuestas en el Capítulo 6.

Parte V: Conclusiones y Trabajos Futuros

Capítulo 10. Conclusiones y Trabajos Futuros. El capítulo se divide en dos bloques claramente definidos. Por una parte, se muestran las conclusiones obtenidas a lo largo del trabajo realizado para alcanzar los objetivos del proyecto y, por otra, se presentan de forma bastante explícita algunos de los caminos a seguir como líneas de trabajo futuro de acuerdo al *Marco* desarrollado.

Parte VI: Apéndices

Se han incluido varios apéndices:

Apéndice A. Presupuesto. Se muestra el presupuesto del proyecto fin de carrera detallado con cada una de las tareas realizadas.

Apéndice B. Instalación y Configuración del Entorno de Pruebas. En este apéndice se detallan cada una de las configuraciones realizadas sobre los equipos que forman el entorno de pruebas, así como se muestra cómo se instalan y configuran algunas aplicaciones necesarias para el testeo del *Marco* de forma rigurosa.

Apéndice C. Manual de Usuario Administrador. Se muestra el manual para configurar los equipos y, por tanto, el sistema de *Difusión Selectiva* desarrollado, a través de las *interfaces gráficas de usuario* que se han implementado sobre el *Marco*. Por tanto, al haberse desarrollado dos sistemas, en este capítulo se muestran estos dos manuales que permiten la configuración de los equipos.

Apéndice D. Instalación de la Aplicación Media Wiki. Se muestra el proceso de instalación de la aplicación *Media Wiki* que se ha utilizado como previsión de generación de una manual en línea por parte del administrador.

Apéndice E. Archivos de políticas URL para aplicaciones SWF. Este apéndice muestra el *archivo de políticas url* definido y utilizado por las *aplicaciones swf* desarrolladas.

PARTE II. Estado del Arte. Soluciones Comerciales

Capítulo 2 Conceptos y Consideraciones de Difusión Selectiva

El objetivo final del proyecto es el de *diseñar y desarrollar un Marco para Aplicaciones de Difusión Selectiva*. Para desarrollar un *Marco* se ha de establecer una base sobre la que comenzar a desarrollar las diferentes aplicaciones. Para ello, inicialmente, se ha de diseñar una arquitectura que provea el funcionamiento de todo el sistema que finalmente se va a desarrollar. La arquitectura en nuestro caso estará formada por diferentes elementos, que definiremos en los sucesivos capítulos del documento y la denominaremos *arquitectura funcional*.

Dentro de las diferentes herramientas analizadas disponibles en el mercado se ha observado que existe ambigüedad para referirse a las aplicaciones de *Difusión Selectiva*; es por ello que, antes de definir una *arquitectura funcional*, se debe establecer un contexto para identificar de forma unívoca los diferentes elementos definidos. Para ello, algunas de las clasificaciones que se introducen en los *Tipos de Enlaces* de este capítulo son utilizadas para establecer analogías y realizar una clasificación muy concreta de los *Tipos de Difusión Selectiva Identificados o Definidos* en el presente trabajo.

También, para centrar el contexto del proyecto fin de carrera, es necesario presentar algunas opciones disponibles para el transporte de la información, en este caso contenido multimedia, para aplicaciones de *Difusión Selectiva*.

Con todo lo anterior, el principal objetivo de éste capítulo es introducir al lector dentro del contexto de la *Difusión Selectiva* para, además de focalizar el problema a tratar, sentar las bases para comprender la propia *arquitectura funcional implementada* y comprender las decisiones de diseño adoptadas sobre ésta en capítulos posteriores.

2.1 Definición Difusión Selectiva o Narrowcasting

La palabra anglosajona, *Narrowcasting*, aún no ha sido introducida en nuestro idioma de forma oficial por la *Real Academia de la Lengua Española*. Sin embargo, la traducción más aceptada por la comunidad de habla hispana que trabaja con estos conceptos la ha denominado *Difusión Selectiva*. Así, algunos movimientos de redes denominadas *Web 2.0*, han realizado una definición más o menos concisa del término. La enciclopedia libre *Wikipedia* [1] define la *Difusión Selectiva o Narrowcasting* [2], con bastante precisión, del siguiente modo:

“Se conoce con el término Narrowcasting o Difusión Selectiva (por oposición a Broadcasting, ‘Divulgación’) a la posibilidad de Difusión Selectiva de publicidad o información hacia segmentos diferenciados por valores, preferencias o atributos demográficos.

El Narrowcasting ha permitido huir del concepto de audiencias masivas. Una de las soluciones actuales dentro de las posibilidades adaptativas del Narrowcasting consiste en la instalación de redes informáticas con ordenadores conectados a monitores o displays que reciben nuevos contenidos publicitarios o informativos cuando se desea su actualización.”

Es recomendable puntualizar que la *Difusión Selectiva (Narrowcasting)* es, por tanto, una herramienta inherentemente creada para realizar *Publicidad Dinámica* o *Cartelería Digital (Digital Signage)* como es denominada por los diferentes fabricantes de *software*. No obstante,

este tipo concreto de *Difusión Selectiva*; es decir, mostrar publicidad mediante contenido multimedia a través de monitores conectados a ordenadores, es un subconjunto menor de todas la posibilidades que ofrecer la *Difusión Selectiva*.

2.1.1 Cartelería Digital (Digital Signage) frente a la Difusión Selectiva (Narrowcasting)

Dentro del contexto de este proyecto se considera como conceptos diferenciados el *Narrowcasting (Difusión Selectiva)* y *Digital Signage (Cartelería Digital)*; no obstante, al estar estrechamente relacionados es común leer artículos que se refieren a ambos términos indistintamente. En el presente trabajo denominaremos *Cartelería Digital*, a la aplicación concreta de la *Difusión Selectiva* que muestra contenido multimedia a través pantallas conectadas a ordenadores o dispositivos con el objetivo de informar o publicitar productos haciendo uso de contenido multimedia. Nos referiremos a la *Difusión Selectiva* cuando se haga referencia a un concepto más general que abarca también otras aplicaciones. Así, siendo más rigurosos y precisos, se debe considerar la *Cartelería Digital o Digital Signage* como un subconjunto del *Narrowcasting o Difusión Selectiva*. De este modo, el *Narrowcasting* representa ciertas funcionalidades de una arquitectura, y el *Digital Signage* es la aplicación de esa arquitectura a un negocio específico: el de la publicidad y, en concreto, la *Publicidad Dinámica*.

Supongamos que se desarrolla una arquitectura a través de aplicaciones y equipos electrónicos que proporciona la *Difusión Selectiva* de contenido multimedia (video, imágenes, animaciones, sonidos). Si se utiliza para presentar publicidad en un centro comercial estaremos hablando de un *Sistema de Difusión Selectiva de Cartelería Digital*. Si, por el contrario, se utiliza ese mismo sistema para proveer *Televisión Digital*, de modo que a ciertos usuarios se le proporciona un contenido y a otros otro en función del tipo de suscripción, se habla de *Sistema de Difusión Selectiva de Televisión*, el cual no puede ser considerado como *Cartelería Digital* ya que el propósito principal no es proveer publicidad, sino contenido. Por norma general, una aplicación de *Televisión Digital* necesitan poder realizar emisiones en directo, mientras que en una aplicación de *Cartelería Digital* basta con proveer la funcionalidad de emitir videos y contenido multimedia bajo demanda, estando, ambas, inmersas en un contexto de *Difusión Selectiva*.

De este modo, aunque se hace una distinción de este término, el presente trabajo pretende que el lector, al final de su lectura, pueda concebir la *Difusión Selectiva*, no sólo como un medio para la *Cartelería Digital*, sino como un nuevo concepto que pueda abarcar nuevas aplicaciones en el campo de la *Ingeniería de Servicios*. Desde el punto de vista del autor deben existir dos objetivos íntimamente relacionados: por una parte, el *desarrollo técnico* del sistema y, por la otra, el *desarrollo del servicio* que se pretende ofrecer, aplicando de forma continua *Ingeniería de Servicios*.

2.1.2 Servicios y Aplicaciones en la Sociedad de la Información

En dispositivos tan relevantes como la *Televisión*, la *Radio*, la *Telefonía o Internet* han ido apareciendo, mediante planteamientos de *Ingeniería de Servicios*, nuevos servicios y aplicaciones diferentes para los que inicialmente fueron concebidos. Incluso ha permanecido el concepto como tal y, sin embargo, la tecnología utilizada ha dado un giro drástico con respecto a la implementación inicial, lo que ha permitido que se añadan nuevas funcionalidades.

Los ejemplos anteriormente mencionados pueden considerarse como la base de la tecnología de las *telecomunicaciones* aplicadas a los servicios y, todas ellas, están enfocadas a proporcionar, desde el punto de vista del autor, los siguientes *favores* a sus usuarios o audiencias:

- *Información*
- *Entretenimiento*
- *Publicidad*
- *Comunicación*

De este modo, partiendo de estos cuatro grandes pilares, se pretende desgranar y presentar, de forma resumida, algunos servicios que se pueden ofrecer mediante un hipotético sistema de *Difusión Selectiva*. Se ha de tener siempre en cuenta, al leer estas líneas, que nuevos servicios están aún por descubrir y que será la imaginación del lector el que, sin duda, pueda proponer o intuir nuevas formas de utilización las herramientas de *Difusión Selectiva*.

A continuación, se presentan algunos de los servicios que se han identificado para hacer uso de este tipo de sistemas:

Centros Comerciales, Supermercados, Establecimientos

Dentro del subconjunto de la *Difusión Selectiva* que se ha denominado *Cartelería Digital*, se hace visible el potencial de utilizar grandes pantallas de presentación de contenido para *informar* de las ofertas del establecimiento en forma de *publicidad*.

Museos, Ferias de Exposiciones

Uno de las aplicaciones más sugeridas en las que se puede utilizar la *Difusión Selectiva* es a la hora de mostrar o *informar* en espacios amplios y situados geográficamente en un emplazamiento o varios. Se incluyen las ferias de exposiciones o cualquier evento con similares características en las que en cada sala o espacio separado se quiere mostrar diferentes contenidos.

Universidades

La universidad es un espacio donde se realiza una enseñanza especializada en materias muy diversas. Además, en general, se tienen diferentes edificios esparcidos geográficamente con usos muy diferentes los unos de los otros, por lo que esta diferenciación también puede ser un motivo importante para presentar contenido desigual: biblioteca, facultades, cafetería, secretaría, instalaciones deportivas,...

La aplicación de un servicio de *Difusión Selectiva* en el ámbito de la universidad puede tener objetivos diferentes: *Informar, Entretener, Publicitar Productos o Comunicar*.

Señal de TV y Canales Corporativos

Nótese, como se ha indicado ya, que hay muchas más aplicaciones que las enfocadas a proporcionar *publicidad*; por lo que la *Difusión Selectiva* puede ofrecer otros servicios diferentes a la *Cartelería Digital*.

Podría utilizarse un servidor para establecer un *Canal de TV* en comunidades de vecinos, de modo que se puede ofrecer contenido multimedia bajo demanda a modo de *Portal Web de Video* o incluso a través de equipos clientes a modo de decodificadores. Así, es posible

distribuir el contenido en forma de señal analógica a través de moduladores que modulan la señal digital de la salida de un *equipo terminal* a una frecuencia determinada *UHF* y ser distribuido a través de un cable coaxial por el cableado de un edificio a un coste relativamente bajo, pudiendo, incluso, ofrecer la difusión de varios canales ante un equipo con varias tarjetas de video.

Podría aplicarse este servicio a una comunidad de vecinos para ofrecer nuevos canales de TV incluyendo publicidad basada en localización, (por ejemplo, tiendas en el barrio), o también se podría generar un *Canal de TV Corporativo* en una empresa sin hacer uso de caros repetidores y nuevo cableado.

Los *canales de TV corporativos* suelen emitir información y publicidad relacionada con la empresa e incluso, a veces, son utilizados para comunicar información importante a los empleados: salidas de emergencia, previsión del tiempo, cambios en la política de la organización,... Son denominados así porque se suelen emitir en las redes internas de las grandes corporaciones.

Medios de Transporte Públicos. Autobuses, Metro

Es sin duda un escenario apropiado para implementar el concepto de *Difusión Selectiva* en la búsqueda de proporcionar las necesidades de *Información, Entretenimiento, Publicidad y Comunicación*.

En las grandes urbes las personas tardan un tiempo considerable en el desplazamiento a sus puestos de trabajo. Son los cuatro grandes pilares mencionados anteriormente (*Información, Entretenimiento, Publicidad y Comunicación*) los que son demandados en estos desplazamientos a través del transporte público. Actualmente, estas necesidades se han provisto a través de publicaciones gratuitas en papel y otros medios más sofisticados como *hardware* específico para el entretenimiento como videojuegos y dispositivos de reproducción de música.

En la actualidad, en algunos medios de transporte, como es el *Metro de Madrid*, se ha desarrollado un *Canal Corporativo de TV* donde se muestran anuncios, noticias, reportajes o las bondades del medio de transporte en cuestión. Esta misma idea se puede trasladar a otros medios de transporte como son los autobuses públicos, los tranvías y trenes. Para el acceso a la red que actualiza el contenido podrían utilizarse tecnologías inalámbricas como *wi-fi* o *WiMAX*.

El contenido, de esta forma, se puede adaptar a las líneas de transporte, ya sea a través de la publicidad (anunciantes de la zona) o incluso del contenido (adaptado al nivel cultural y costumbres). También, incluso la difusión del contenido en estas plataformas podría adaptarse a la situación geográfica de los medios de transporte, con la adquisición de información mediante sensores, por ejemplo, *GPS*, temperatura,...

Internet y Otros Medios

Mediante el uso de *Internet* y las *redes internas (Intranets)* de las corporaciones se pueden ofrecer otras aplicaciones en las que se hagan uso de *Difusión Selectiva*.

Es posible establecer aplicaciones de *Difusión Selectiva* como un *Portal Web de Video* que proporcionen contenido multimedia en función de criterios tales como el perfil del usuario que accede al sistema o *cookies* almacenadas en el navegador que permiten diferenciar las búsquedas realizadas con anterioridad.

2.2 Tipos de Enlaces de Comunicación Básicos

En el presente *proyecto fin de carrera* se va a realizar un pequeño aporte en definiciones en sub-apartados posteriores (apdo. 2.5 Definición de *Tipos de Difusión Selectiva*) apoyándonos, en parte, en analogías con los *Tipos de Enlaces de Comunicación* presentados a continuación.

Dentro de las redes y conexiones que se pueden establecer, los enlaces que interconectan los *nodos* de una red se pueden clasificar en tres tipos según el sentido de las comunicaciones que transportan:

- **Simplex:** La transacción sólo se efectúa en un solo sentido.
- **Half-Dúplex:** La transacción se realiza en ambos sentidos, pero de forma alternativa, es decir solo uno puede transmitir en un momento dado, no pudiendo transmitir los dos al mismo tiempo.
- **Full-Dúplex:** La transacción se puede llevar a cabo en ambos sentidos simultáneamente.

Cuando la velocidad de los enlaces *Half-Dúplex* y *Dúplex* es la misma en ambos sentidos, se dice que es un *enlace simétrico*, en caso contrario se dice que es un *enlace asimétrico*.

Esta clasificación nos servirán posteriormente para establecer definiciones de los *Tipos de Difusión Selectiva* que se han detectado y definido atendiendo a la forma de realizar la transacción de información entre los diferentes entes.

2.3 Difusión Selectiva (Narrowcast) y Comunicaciones (Unicast, Broadcast y Multicast)

En terminología de redes y comunicaciones hay que tener claro la diferencia entre los conceptos relacionados con el envío de paquetes como son *Unicast (Unidifusión)*, *Multicast (Multidifusión)* y *Broadcast (Divulgación)* de modo que se podrán establecer sistemas *Narrowcast (Difusión Selectiva)* con diferentes estrategias de envío de la información.

Estas definiciones, por un lado permiten clasificar las redes y sus protocolos en función de los participantes en una comunicación, y por otro son muy útiles para definir esa misma característica en aplicaciones que utilizan la red.

A continuación, se presentan las definiciones de comunicaciones *Unicast*, *Multicast* y *Broadcast* establecidas en el *proyecto fin de carrera*:

- **Unicast:** Se basa en un proceso de envío de una información en una o más unidades de datos (datagramas IP [3]) desde un equipo origen a un único equipo destinatario o receptor final. Si se desea enviar la misma información y hay n destinatarios, habrá n comunicaciones independientes o n copias de la misma información enviadas desde el equipo origen.

A continuación, se muestra un ejemplo consistente en tres comunicaciones *Unicast*:

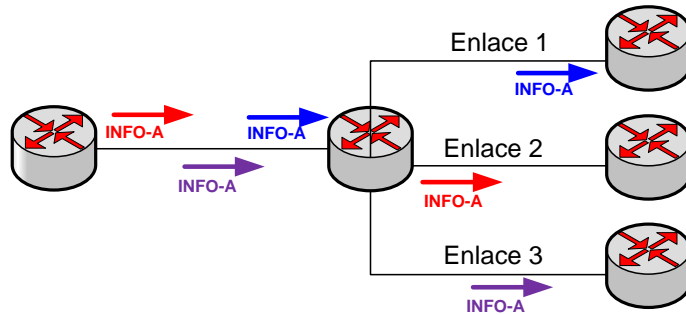


Ilustración 1.-. **Unicast**. La información se transmite hacia los nodos de forma independiente en conexiones diferentes

- **Multicast:** Se basa en un único proceso de envío, independientemente del número de potenciales equipos receptores. Se envía una misma información en una o más unidades de datos (datagramas IP) desde un equipo origen a todos los nodos destinatarios [4] que posean al menos un miembro de un determinado grupo multidifusión. Además deben compartir una misma dirección de multidifusión y, posiblemente, los equipos se encuentren dispersos geográficamente en múltiples redes. Se resalta el hecho de que desde la máquina origen sólo se envía una vez la pertinente información y no se transmiten n copias de la misma aunque haya n destinatarios. En este escenario, los routers intermedios de multidifusión de la red tienen que poseer previamente la capacidad necesaria para hacer las copias de la información transmitida a cada enlace.

A continuación (Ilustración 2), se muestra una *red punto multipunto* en la que se implementa una comunicación *Multicast*:

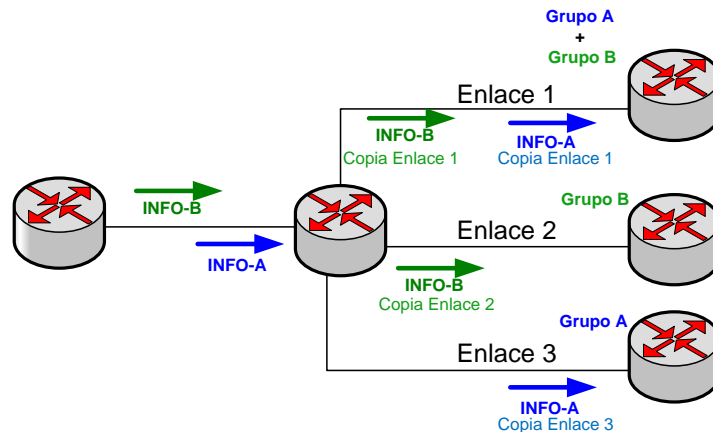


Ilustración 2.-. **Multicast**. La información se transmite hacia los nodos que pertenecen al grupo de multidifusión de la información enviada.

- **Broadcast:** Es una forma de enviar la información donde el equipo origen envía la información a todos los equipos destinatarios de manera simultánea. De este modo, no es necesario repetir la misma comunicación nodo por nodo [5]. En un sistema *Broadcast* se emite una señal que puede ser recibida por cualquier dispositivo que cuente con un receptor apropiado. El nombre de *Broadcast* (*Divulgación*) se origina en la radio y la televisión, y es característico de estos sistemas de difusión.

A continuación (Ilustración 3), se muestra una *red punto a multipunto* en la que se implementa una comunicación *Broadcast*:

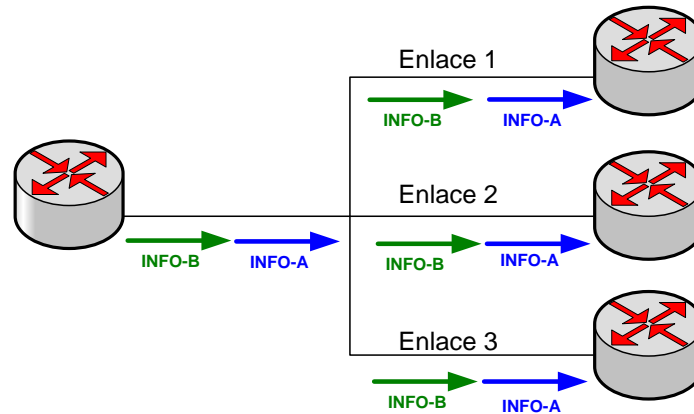


Ilustración 3.-. Broadcast. La información se envía directamente a todos los enlaces.

Los tres tipos de comunicaciones nos pueden permitir construir sistemas *Narrowcast*, vamos ahora a ver cómo. Es posible establecer un sistema *Narrowcast* mediante comunicaciones *Unicast*. Así, la información enviada a cada dispositivo final dependerá, por ejemplo, del perfil del receptor. Un ejemplo muy interesante de cómo se establece *Narrowcast* mediante comunicaciones *Unicast* se está llevando a cabo continuamente en la publicidad que recibimos cuando navegamos por *Internet*. La publicidad recibida se establece en función del contenido consultado y es enviado a través de conexiones *Unicast*. Diferentes implementaciones de TV sobre IP [6] permiten también recibir contenido *Narrowcast* mediante conexiones *Unicast*, por ejemplo, en formatos de pago por visión. Existen portales de video que establecen directamente el contenido mediante una conexión *Unicast* en función del perfil configurado o *cookies* [7] almacenadas por el navegador, por lo que podría considerarse también como un sistema *Narrowcast*.

Un ejemplo clásico de *Multicast* puede ubicarse en la televisión por cable [8], en la que no se emplean ondas electromagnéticas en la atmósfera o en el espacio como medio de transmisión, sino las que se pueden establecer en el cable. En este caso, la información solamente llegará a los suscriptores conectados al cable (y abonados al grupo de multidifusión). Además, es posible enviar distinta información en función de si el abonado está suscrito a determinados canales. Conceptualmente establecer *Multicast* en una red permite establecer sistemas *Narrowcast*, debido a que puede diferenciarse la recepción del contenido atendiendo a los grupos a los que pertenece cada dispositivo final.

Por último, en relación al *Broadcast*, usualmente se ha transmitido una señal en abierto; es decir, capaz de activar cualquier dispositivo dotado del sintonizador apropiado. Este es el caso de la radio y la televisión tradicionales. Pero ya es muy común también la alternativa en que se transmite en *Broadcast* una señal codificada de tal manera que no baste un sintonizador públicamente disponible para convertirla en datos, sonido o imágenes, sino que se requiera un decodificador que debe ser adquirido o alquilado por separado del equipo receptor. Así, han ido apareciendo sistemas más sofisticados con el objetivo de obtener sistemas conceptualmente *Narrowcast*, a través de una emisión *Broadcast* de contenido codificado. Otro ejemplo, lo encontramos en la televisión restringida a través de satélite que conceptualmente es una forma de *Narrowcast* basada en *Broadcast*, ya que limita el contenido a aspectos relacionados con el perfil del usuario o *público objetivo*.

2.4 Tecnologías de Red para Transportar Tráfico de Aplicaciones de Difusión Selectiva

Uno de los aspectos críticos en un despliegue de red acertado es establecer una armonía cuidadosa entre los *servicios* que se proporciona y la disponibilidad de *ancho de banda*. Archivos de alta calidad, como aquellos que se emitirán en el *Cartelería Digital*, son intrínsecamente ‘pesados’ (grandes en uso de memoria física y, por tanto, utilización de gran *ancho de banda* en su transporte) y pueden causar problemas de congestión serios si no son manejados correctamente. Así, la evaluación apropiada inicial de la solución necesaria y la planificación pueden hacer ahorrar costes y dinero a la organización: prevención de productividad perdida, funcionamiento pobre de red perjudicando otros servicios (*ToIP*, *Videoconferencias*,...) y usuarios descontentos, por poner algunos ejemplos.

Por estos motivos, es importante evaluar y tener en cuenta las diferentes tecnologías de red para la distribución del contenido multimedia a la hora de elegir la mejor solución posible. Este proyecto no sólo se centra en las soluciones más básicas presentes en **Redes de Distribución de Área Local** (LAN, *Local Area Network*), sino que también se presentan las **Redes de Distribución de Área Amplia** (WAN, *Wide Area Network*), donde deberían entrar *arquitecturas funcionales* que al menos presenten características de uso eficiente del *ancho de banda*.

2.4.1 Redes de Distribución de Área Amplia (WAN)

Internet por cable y xDSL

En muchos de los negocios de hoy en día, *Internet* se ha convertido en parte de la arquitectura de red de las organizaciones. Muchas empresas, tanto grandes como pequeñas, instalan y configuran *VPNs* (*Virtual Private Networks* o en castellano *Redes Privadas Virtuales*) sobre los servicios de *banda ancha* de los operadores, para así ahorrar costes en comparación con el hecho de alquilar líneas dedicadas mediante tecnologías *Frame Relay*, *MPLS* u otras tecnologías [8]. La flexibilidad y el ahorro que supone representan una compensación potencial frente a la posibilidad de comprometer los niveles de servicio.

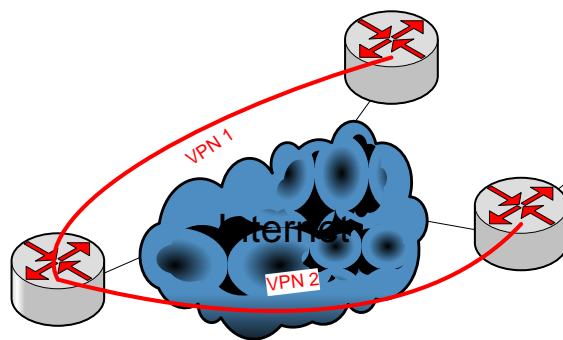


Ilustración 4.- Esquema básico de sedes interconectadas a través de Internet mediante VPNs.

Como es sabido, los *Proveedores de Servicio de Internet* (*ISP*, *Internet Service Provider*), a menudo, reducen el *ancho de banda* total en el final del acceso al cliente, para así suministrar enlaces a más usuarios sobre una misma línea troncal. Se debe tener presente siempre esta limitación cuando se pretende establecer *Streaming* [9] de contenido multimedia para *Difusión Selectiva*, sobre una conexión que da acceso libre a Internet. Es decir, el *ancho de banda* de la línea se puede reducir considerablemente si los usuarios del servicio colindante comienzan a hacer uso de sus conexiones todos de forma simultánea.

Red Privada Dedicada o Arrendada

Las *Líneas Privadas Arrendadas*, ya sean establecidas mediante *Frame Relay*, *ATM* [10] [11] o cualquier otra tecnología actual, como *MPLS*, han sido los cimientos de conectividad de las empresas durante años. En los inicios del mercado de las redes de comunicación era el único método que tenían las empresas para interconectar sus redes de datos. Con las *Líneas Privadas Arrendadas* las empresas manejan y deciden sus propios flujos de datos. Por tanto, los *departamentos de comunicaciones*, con este tipo de enlaces, observan y analizan sus necesidades supervisando los flujos de datos y manteniendo su canal contratado justo en el límite de saturación. De este modo, pueden decidir reducir o aumentar la velocidad de *Tasa Mínima Garantizada (CIR, Committed Information Rate)* que soporta el enlace contratado a su *Proveedor de Servicio (ISP, Internet Service Provider)* [12].

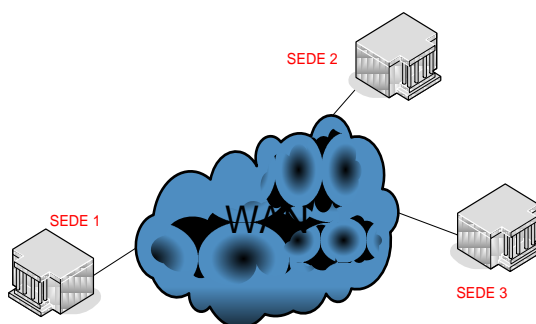


Ilustración 5.- Esquema básico de interconexión de sedes a través de Líneas Privadas Dedicadas.

Así, instalando una red para aplicaciones de *Difusión Selectiva*, se debería supervisar exhaustivamente el uso del *CIR*. Mediante este control se deberá asegurar que el flujo de contenido del material multimedia derivado de la arquitectura implementada no afecta al flujo de tráfico normal de la red. Las *Líneas Privadas Dedicadas o Arrendadas*, pueden ser la mejor opción para una *Red de Difusión Selectiva* que realiza *Streaming*, por ejemplo, pero es también la más costosa. Se puede obtener un *retorno de la inversión* bueno de las redes utilizando aplicaciones de *Difusión Selectiva*, sin embargo, hay que asegurarse que el *ancho de banda* disponible se utiliza al máximo.

Satélite

Los satélites han sido usados durante años como un método de distribución de video. La mayoría de los enlaces vía satélite se establecen de forma *Multicast/Broadcast* a grupos geográficos y han sido una buena forma de distribución para el video. Por ejemplo, la TV por satélite ha utilizado el formato de video *MPEG-2*, el cual es uno de los formatos más cotidianos en sistemas de *Difusión Selectiva*.

En el pasado, el uso de tecnologías *IP* sobre satélite, estaba limitado debido a la latencia en el tratamiento y transporte de datos por satélite; sin embargo, los satélites de hoy en día pueden transmitir y recibir paquetes *IP* en ida y vuelta de aproximadamente entre *500 mseg* y *600 mseg* [13].

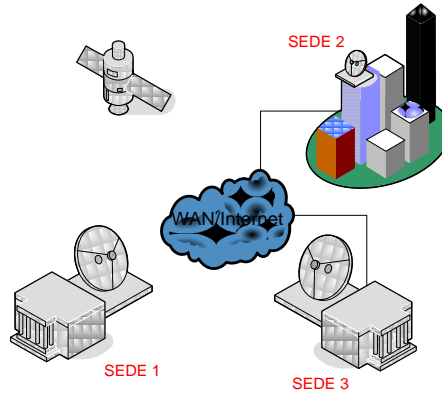


Ilustración 6.-. Esquema básico de utilización de enlaces satélite. Se observa en el esquema la posibilidad de conectar sedes aisladas.

Muchas organizaciones utilizan enlaces de satélite como *backup* o quedan reservados ante fallos de los enlaces terrestres. Es habitual para los enlaces satélites que queden infrautilizados si no se producen fallos en los enlaces principales terrestres. Algunos fabricantes han diseñado módulos para enviar datos IP a través de satélites; así, con estas nuevas tecnologías, se pueden usar los enlaces vía satélite para encaminar los datos pertenecientes a la *Difusión Selectiva* de la organización. Otra ventaja de utilizar satélites para el transporte de datos, es que la localización geográfica queda independiente de la conexión con enlaces terrestres.

Otras Tecnologías Emergentes

Además de las tecnologías expuestas están surgiendo otras nuevas con características innovadoras y mejoradas con respecto a las presentadas. Estas tecnologías también pueden proporcionar *Redes de Distribución de Área Amplia (WAN)*, pero con características inalámbricas y, aunque su uso aún no ha quedado ampliamente utilizado, son de consideración debido a la inminente implantación de algunas de estas redes, a veces denominadas redes de cuarta generación [14].

La tecnología más representativa podría ser la tecnología *WiMAX* [15], la cual permite el acceso a la red a una amplia zona geográfica a partir de un mismo nodo.

Es destacable que estas tecnologías son importantes para el desarrollo de zonas rurales donde la población queda dispersa y resulta costoso desplazar enlaces terrestres y pueden ser muy ventajosas a la hora de distribuir contenido multimedia a través de arquitecturas de *Difusión Selectiva*.

2.4.2 Redes de Distribución de Área Local (LAN)

Red Cableada

En la mayoría de las situaciones, la configuración óptima requiere un equipo terminal, para presentar los contenidos multimedia programados, el cual se puede conectar directamente a un punto de acceso cableado de la *Red de Área Local*.

Las redes cableadas proveen a las organizaciones un medio de distribución confiable, eficiente y de *banda ancha*, permitiendo diferentes tipos de flujos en la *LAN* con los mínimos efectos sobre el resto de flujos. Una red cableada bien configurada será la solución óptima para conectar los *equipos terminales* y el *servidor de contenido* de la *arquitectura funcional* que se ha desarrollado.

Red Inalámbrica

Se ha de pensar, desde el punto de vista de la ingeniería, en una solución inalámbrica si una solución cableada no es posible. La combinación de un punto de acceso inalámbrico y un equipo cliente con capacidades inalámbricas (*tarjeta wi-fi*), provee una solución con suficientes garantías en un entorno normal de trabajo.

Una red vía enlace radio; es decir, una red inalámbrica, permite tener acceso a lugares donde la *Difusión Selectiva* podría ser eficaz; pero, sin embargo, queda fuera del alcance del cableado físico. Por ejemplo, en un sistema de *Difusión Selectiva* en una tienda (*Cartelería Digital*), el equipo terminal se suele colocar en el escaparate o a la entrada para que así los potenciales clientes tengan un impacto sobre los productos que ofrece el establecimiento. Sin embargo, en muchas ocasiones, el conjunto formado por equipo terminal y el equipo de presentación (*TFT, Monitor, TV,...* además de soportes) suele ser móvil, por lo que así se evitan las molestias de cables.

Muchas veces, por tanto, puede llegar a ser una mejor solución el acceso al medio a través puntos de acceso inalámbricos. No obstante, también se ha de indicar que es posible que el equipo servidor de contenido deba residir bajo un ambiente de *Red Cableada*, fundamentalmente por temas relacionados con la seguridad y la prevención.

2.5 Definición de Tipos de Difusión Selectiva

La *Difusión Selectiva* permite emitir contenido diferenciado según sean las características del *público objetivo*. Así, es posible programar las *listas de reproducción* en función de valores, preferencias o atributos geográficos de este *público objetivo*.

Cuando seleccionamos un contenido distinto para dos emplazamientos físicos diferentes según, por ejemplo, el idioma habitual de la región o el nivel adquisitivo de *público objetivo* estamos realizando *Difusión Selectiva*. Además, con las nuevas tecnologías es posible diferenciar y clasificar el *público objetivo* con una red de sensores en cada emplazamiento físico y adecuar el contenido a presentar en función de *público real* (público que se encuentra en un instante delante del escenario) y no el *público potencial* o *público objetivo* (público en general que se espera tener en media en un determinado escenario). También es posible incluso esperar la colaboración del *público real* habilitando la posibilidad de auto-configurar las listas de reproducción a través de la interactividad con los ciudadanos que se encuentran en un emplazamiento en un momento determinado.

En este apartado se presenta una clasificación de los tipos de *Difusión Selectiva* identificados en función de cómo el *dispositivo terminal* establece la comunicación con el *público real*. Esta clasificación no es arbitraria y se ha establecido según las actuales tendencias comerciales, las cuales se resumirán en el Soluciones para Proporcionar Servicios de Difusión Selectiva.

En los siguientes sub apartados, para establecer la clasificación, haremos uso de los conceptos introducidos hasta ahora, pero aplicados dentro del ámbito de la *Difusión Selectiva*.

2.5.1 Difusión Selectiva Pasiva

Esta modalidad de *Difusión Selectiva* considerada es la más sencilla de todas las posibles ya que no incluye comunicación del equipo terminal con el *público objetivo* a través de una potencial red de sensores. Así, el administrador del sistema programa el contenido que se emitirá en cada una de los emplazamientos físicos en función de la *lista de reproducción* fija, la

cual no se adapta a las señales que pudieran ser captadas a través de una hipotética red de sensores.

De este modo se establece una comunicación **Simplex** entre el *equipo terminal* y el *público objetivo*; donde es el equipo terminal el que envía la información a través de contenido multimedia, al *público objetivo*, mediante pantallas y otros medios de difusión.

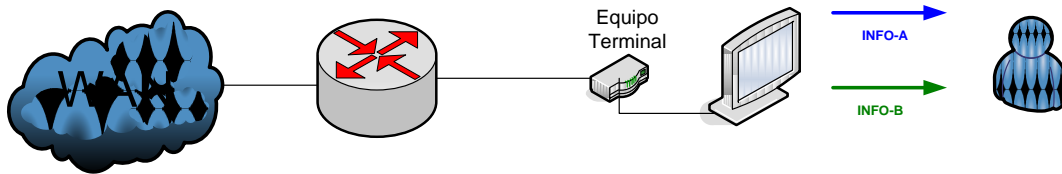


Ilustración 7.-. Difusión Selectiva Pasiva: La información solamente va en un sentido, el contenido no se adapta al entorno.

Como se observa en la ilustración anterior, no existe ningún tipo de comunicación del *público real* hacia el *equipo terminal*, por lo que la *lista de reproducción* no puede ser modificada según la cantidad o el perfil de las personas que se encuentren delante de la pantalla de presentación de contenido.

2.5.2 Difusión Selectiva Adaptativa o Activa

Este tipo de *Difusión Selectiva* se añade un nuevo escalón de complejidad a la arquitectura que la implemente. De hecho, añade necesariamente nuevos dispositivos electrónicos al sistema, ya que se ha de tener en cuenta el estado de los sensores que se integren dentro de la solución. Estos sensores pueden ser de características muy variadas, sin embargo, todos intentan caracterizar de alguna forma el estado del *público objetivo* intentando medir ciertas características del *público real*.

Estos dispositivos podrían ser: cámaras infrarrojas, webcams, sensores de movimiento, micrófonos, detectores de presencia, *bluetooth* y *wi-fi* (en detectores de dispositivos portátiles para adquisiciones de perfiles)... Con estas posibilidades vemos que el fin de estos detectores podría ser muy variado, desde intentar ahorrar energía cuando no haya *público real*, hasta intentar discernir el sexo de la persona que se presenta delante de la pantalla de presentación de contenido.

Un aspecto común a este tipo de *Difusión Selectiva* es el hecho de que estas arquitecturas modifican en tiempo real las *listas de reproducción de contenido* y por tanto la información presentada. Se establece una comunicación **Half-Duplex**, desde el *público real* hacia el *equipo terminal* y posteriormente del *equipo terminal* al *público real*.

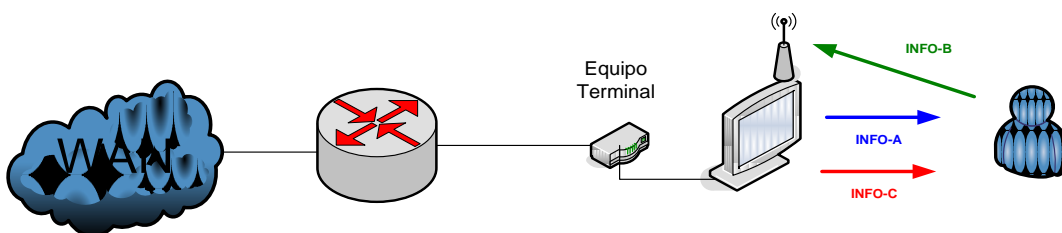


Ilustración 8.-. Difusión Selectiva Activa: La información mostrada por el dispositivo de presentación se adapta al entorno detectado.

Se ha elegido el término activa, por la capacidad del sistema en modificar su comportamiento en función de la actividad detectada del exterior, estableciendo un comportamiento activo con el medio.

2.5.3 Difusión Selectiva Interactiva

Este tipo de difusión es quizás el tipo de *Difusión Selectiva* que pueda captar con mayor éxito la atención del *público real*.

Diferentes modelos pueden proponer diferentes tipos de interactividad con el *público objetivo* y se pueden, del mismo modo que en la *Difusión Selectiva Activa*, utilizar diferentes tipos de dispositivos electrónicos: pantallas táctiles, sensores de movimiento, *bluetooth*, infrarrojos, *wi-fi*,...

En este tipo de intercambio de información la *lista de reproducción* no varía en función del estado de los sensores sino directamente a través de la respuesta que dé el *público real*. Incluso estos sistemas pueden llegar a establecer la posibilidad de que el *equipo terminal* pregunte directamente cuales son los contenidos de interés y, según los datos recibidos, establecer la *lista de reproducción* de forma totalmente dinámica huyendo de patrones preestablecidos. Este tipo de comunicación puede considerarse **Full-Duplex** si se considerara el *público real* y el *equipo terminal* como la analogía de un enlace, debido a que ambos pueden estar enviando información en el mismo instante.

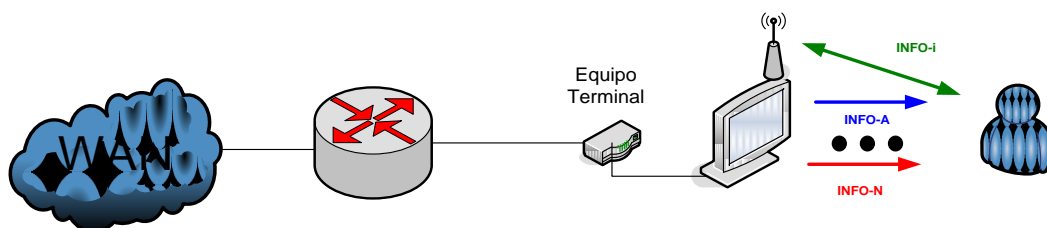


Ilustración 9.-. Difusión Selectiva Interactiva: El usuario intercambia continuamente información con algún dispositivo sensor; así la información mostrada cambia continuamente.

Se podrá diferenciar la *Difusión Selectiva Interactiva* de la *Difusión Selectiva Activa*, en el hecho de que la *Difusión Selectiva Activa* genera las *listas de reproducción* instantáneas a partir de estados preconcebidos a priori, mientras que las *listas de reproducción* en *Difusión Selectiva Interactiva* se va configurando a partir de un cierto algoritmo cuyas variables de entrada son las respuestas que vaya dando el interlocutor (*público real*) hacia los sensores disponibles (*equipo terminal*). Un caso destacado de la *Difusión Selectiva Interactiva* es la que adopta esta interacción a través de juegos entre varios participantes, los cuales pueden manejar diferentes elementos a través de dispositivos móviles con conexiones inalámbricas como son el *bluetooth* o *wi-fi*.

2.6 Transporte del Contenido Multimedia en Difusión Selectiva

El modo de transportar el contenido multimedia tiene una gran importancia, tanto a la hora de configurar la red, como a la hora de desarrollar las aplicaciones que presentarán dicho contenido. Ambas deben estar relacionadas, ya que no es posible, por ejemplo, reproducir un flujo *Multicast* en una aplicación, si este no se ha configurado correctamente en la propia red. Aún así, existen varios mecanismos para reproducir el contenido multimedia, ya que existen protocolos para transportar el contenido con diferentes métodos ya sea mediante envío de los archivos en forma *Broadcast*, *Unicast* o *Multicast*.

En este apartado se pretende resumir las alternativas disponibles de forma general, para posteriormente establecer una elección en la implementación del *Marco*.

2.6.1 *Descarga y Reproducción, Pseudo Streaming (Descarga Progresiva) y Streaming Media*

Existen tres métodos para entregar datos de audio y video a través de una red: por *Descarga y Reproducción*; por *Descarga Progresiva (Pseudo Streaming)* [16] [17] y por *Streaming (Streaming Media)* [18]. Con la intención de diferenciar cada tecnología los proveedores de las mismas han hecho su propio argot. Hay fabricantes de software que se refieren a la *Descarga Progresiva (Pseudo Streaming)* como simplemente *Streaming*. En este apartado se va a puntualizar algunos de los términos, para posteriormente seleccionar la mejor opción, de las consideradas, para el propósito de *proyecto fin de carrera*.

Se define *Streaming* como una corriente continua de datos sin interrupción [19] [20]. Otras definiciones, consideran el *Streaming* como el término que se refiere a la *entrega en tiempo real* de audio y video a través de una red comunicaciones, en algún caso denominado de forma específica como *Live Streaming* [18]. En todos los casos, este tipo de tecnología permite que se almacene en un *búfer* lo que se va escuchando o viendo. El *Streaming*, considerado de forma genérica, ofrece, por tanto, la posibilidad escuchar música o ver videos sin necesidad de ser descargados previamente.

A continuación, se hace una definición más precisa de los métodos considerados para la entrega de datos de video y audio. El objetivo de estas definiciones es el de eliminar las ambigüedades en el contexto de *proyecto fin de carrera*:

- ***Descarga y Reproducción:*** Para aplicar este método de reproducción, primero hay que descargar completamente el archivo de audio o video de forma local antes de poder reproducirlo. Por esta razón, la *Descarga y Reproducción* no puede ser utilizada como una forma de transmisión en vivo, ni tampoco como un mecanismo aceptable de reproducción de contenido si existen *listas de reproducción variables* o el contenido de estas listas es modificable, como es el caso de la *Difusión Selectiva*. En ningún caso este método de reproducción puede considerarse como *Streaming*.
- ***Descarga Progresiva o Pseudo Streaming:*** Este particular mecanismo, a veces también denominado simplemente *Streaming* (o *HTTP Streaming* según el protocolo más utilizado para llevarla a cabo) [17], permite reproducir el archivo audiovisual mientras se va descargando al dispositivo de forma local. Este método tiene como principal ventaja el hecho de que no hay que esperar a que termine la descarga para comenzar la reproducción. La *Descarga Progresiva o Pseudo Streaming* es muy útil y totalmente factible para ser utilizada en ciertas aplicaciones de *Difusión Selectiva*. Existen algunas limitaciones: por ejemplo, si el archivo no se descarga tan rápido como se reproduce, entonces se observará que la reproducción se detiene mientras esperas para que se descargue progresivamente el archivo. Esto puede ser un poco incómodo y es por lo general la principal razón por la cual algunas personas prefieren el *Streaming Media* para disfrutar de los contenidos audiovisuales. Esta característica hace muy difícil emitir contenido en directo mediante estrategias de *Descarga Progresiva*.
- ***Streaming Media:*** Permite visualizar y/o escuchar en tiempo real aquello que se ofrece en la red. Por tanto, el *Streaming Media* permite también la transmisión en vivo de audio y video, transformando la red en un gigantesco transmisor de medios audiovisuales. Así, en principio, el *Streaming Media* es la única variedad de *Streaming* que posibilita la transmisión en directo.

Por tanto, para el transporte del contenido multimedia en aplicaciones de *Difusión Selectiva* se tienen dos alternativas: *Pseudo Streaming* y *Streaming Media*. En aplicaciones prácticas la diferencia entre la *Pseudo Streaming* y el *Streaming Media* no siempre es obvia, ya que si el *ancho de banda* para reproducir el video es suficiente apenas se observarán diferencias en la reproducción. No obstante, si el *ancho de banda* es deficitario en el *Streaming Media* la línea de tiempo no se verá afectada, aunque la calidad del video se verá degradada por la pérdida de paquetes, incluso llegando a perder totalmente la imagen. Por otra parte, en el modo *Pseudo Streaming*, el video (o audio) no sufrirá pérdida de paquetes y se parará hasta ir obteniendo los paquetes para su completa reproducción; sin embargo, la duración de la reproducción del video se verá afectada. Estas diferencias aparecen precisamente por la forma de enviar el tráfico sobre la red; generalmente el *Pseudo Streaming* lo hace mediante el protocolo *TCP* (incluso *http* en las capa más altas) y el *Streaming Media* lo ha hecho tradicionalmente mediante *UDP* (aunque también permite *TCP* en ciertos protocolos).

2.6.2 Alternativas al Transporte del Contenido Multimedia

Existen varios mecanismos de transporte del contenido multimedia. En este sub apartado se van a enumerar algunos de los protocolos y estándares ampliamente utilizados para el envío y recepción del contenido multimedia. Se ha de tener en cuenta que se establecerá un pequeño resumen, sin entrar en detalles técnicos de la implementación de cada estándar o protocolo.

A *nivel de sesión* se tienen como más significativos los siguientes protocolos:

- **RTP** (*Real-time Transport Protocol*): *Protocolo de Transporte de Tiempo Real*. Es un protocolo de *nivel de sesión* utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en una *videoconferencia*. Inicialmente se publicó como protocolo *Multicast*, aunque se ha usado en varias aplicaciones *Unicast*. Se usa frecuentemente en sistemas de *Streaming Media* junto a *RTSP*, para *videoconferencia* y sistemas *push to talk* y representa también la base de la industria de *VoIP*.

Está desarrollado por el grupo de trabajo de transporte de audio y video del *IETF*. La *RFC 1890* [21], sustituida por la *RFC 3551* [22] (conocida como *STD 65*), define un perfil para conferencias de audio y vídeo con control mínimo. El protocolo se complementa con *RTCP* (*RTP Control Protocol*) y se sitúa sobre *UDP* en el modelo *OSI*; sin embargo, el *RFC 5761* [23] actualiza ciertos aspectos relacionados con la multiplexación de *RTP* y *RTCP* sobre el mismo puerto *UDP*.

- **SRTP** (*Secure Real-time Transport Protocol*): define un perfil de *RTP* con la intención de proporcionar cifrado, autenticación del mensaje, integridad y protección contra reenvíos a los datos *RTP* en aplicaciones *Unicast* y *Multicast*. Dado que *RTP* está muy relacionado con *RTCP*, el cual se usa para controlar una sesión *RTP*, *SRTP* también tiene un protocolo complementario llamado *Secure RTCP* (o *SRTCP*). *SRTCP* proporciona las mismas características relacionadas con la seguridad a *RTCP*, al igual que hace *SRTP* con *RTP*. El empleo de *SRTP* o *SRTCP* es opcional al empleo de *RTP* o *RTCP*; pero incluso utilizando *SRTP/SRTCP*, todas las características que estos protocolos proporcionan (tales como cifrado y autenticación) son opcionales y pueden ser habilitadas o deshabilitadas por separado. La única excepción a esto último es la autenticación de los mensajes, que es obligatoria cuando se está usando *SRTCP*.

Así, la RFC 3711 [24] define *SRTP (Secure Real-time Transport Protocol)*, una extensión del perfil de *RTP* para conferencias de audio y vídeo que puede usarse opcionalmente para proporcionar confidencialidad, autenticación de mensajes y protección de reenvío para flujos de audio y vídeo.

A nivel de aplicación también existen protocolos que condicionan el transporte del contenido multimedia:

- **RTSP (Real Time Streaming Protocol):** Es protocolo a nivel de aplicación que permite flujos de datos en tiempo real. Establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de vídeo. *RTSP* es un protocolo no orientado a conexión, en lugar de esto el servidor mantiene una sesión asociada a un identificador, en la mayoría de los casos *RTSP* usa *TCP* para datos de control del reproductor y *UDP* para los datos de audio y vídeo aunque también puede usar *TCP* en caso de que sea necesario.

La RFC 2326 [25] describe el protocolo *RTSP*. En el transcurso de una sesión *RTSP*, un cliente puede abrir y cerrar varias conexiones de transporte hacia el servidor para satisfacer las necesidades del protocolo. De forma intencionada, el protocolo es similar en sintaxis y operación a *HTTP* de forma que los mecanismos de expansión añadidos a *HTTP* pueden, en muchos casos, añadirse a *RTSP*. Actualmente el borrador, *draft-ietf-mmusic-rfc2326bis-26* [26], actualiza *RTSP* a *RTSP v2* el cual se está elaborando para establecer las necesidades actuales de *Streaming Media* en tiempo real.

- **HTTP (HyperText Transfer Protocol):** el *Protocolo de Transferencia de Hipertexto* es el protocolo usado en cada transacción de la *World Wide Web*. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como *User Agent*. A la información transmitida se la llama *recurso* y se la identifica mediante un *localizador uniforme de recursos (URL, Uniform Resource Locator)* [27]. Los recursos pueden ser archivos, videos, audio o el resultado de la ejecución de un programa, una consulta a una base de datos,... Esta flexibilidad en las funcionalidades de transporte de recursos del protocolo lo convierten en apropiado para transportar contenido multimedia.

La publicación de una serie de *RFCs*, el más importante de ellos es la RFC 2616 [28] que especifica la *versión 1.1. HTTP*, define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (*clientes, servidores, proxies*) para comunicarse.

- **RTMP (Real Time Messaging Protocol):** el *Protocolo de Mensajería en Tiempo Real* es un protocolo propietario desarrollado por *Adobe Systems* para el *Streaming Media* de audio, vídeo y datos a través de *Internet*[29], entre un reproductor de *Flash* [30] (*Flash Player, Flash Air*) y un servidor que implemente el protocolo. Existen diversos servidores que implementan *RTMP*, entre ellos: *Flash Media Server*[31], *Helix Universal Server* [32], *Wowza Media Server*[33], *Red5*[34],...

El protocolo *RTMP* está publicado como especificación abierta [29].

La elección de los diferentes protocolos, sin duda condicionará las características del tipo de transporte establecido para el contenido multimedia: *Pseudo Streaming* (en su forma *HTTP*

Streaming) o *Streaming Media*. Así, se puede intuir que el *Streaming Media* hará uso de *UDP* (también *TCP*) y *RTP (SRTP)*, *RTSP* o *RTMP*, mientras que un mecanismo *Pseudo Streaming* generalmente utilizará *TCP* y *HTTP* para el transporte del contenido.

2.6.3 Codificación del Contenido Multimedia. Contenedores Multimedia.

Establecer cuál es el mejor método de codificación del contenido multimedia es complejo y depende de la aplicación y las necesidades para la cual se pretenda difundir dicho contenido.

Códec es la abreviatura de codificador-decodificador. Describe una especificación desarrollada en *software*, *hardware* o una combinación de ambos, capaz de transformar un archivo de un flujo de datos o una señal [35]. Los *codecs* pueden codificar el flujo o la señal y recuperarlo o descifrarlo del mismo modo para la reproducción o la manipulación en un formato más apropiado para estas operaciones. De este modo los archivos multimedia deberán usar algún tipo de *códec* para almacenar, enviar y reproducir los datos. Un *códec* de video, imagen o audio es un tipo de *códec* que permite comprimir y descomprimir video, imágenes o audio digital. Normalmente los algoritmos de compresión empleados conllevan una pérdida de información.

Así, existe un complicado equilibrio entre mucho de los factores que los definen, entre ellos destacan:

- Calidad de video y/o audio
- Cantidad de datos necesarios para representarlo (*tasa de bits* o *bitrate*)
- Complejidad (computacional) de los algoritmos de codificación y decodificación
- Robustez frente a las pérdidas de datos y errores
- Facilidad de edición
- Posibilidad de acceder directamente a ciertas posiciones de la reproducción

Por tanto según la aplicación para las que han sido diseñados se cuidará con diferente detalle cada una de las características, lo que ha hecho que surja una gran variedad.

A continuación (Tabla 1), se muestra un resumen con algunos *códec* de video, audio e imágenes:

Codificación	MPEG	ISO/IEC	ITU/T	Otros
Compresión de Video	MPEG-1 MPEG-2 MPEG-4 MPEG-4/AVC		H.261 H.262 H.263 H.264	AVS Dirac Indeo MJPEG RealVideo Theora VP7 DivX WMV
Compresión de Audio	MPEG-1 Audio Layer 3 (MP3) MPEG-1 Audio Layer 2 (MP2) AAC HE-AAC		G.711 G.722 G.722.1 G.722.2 G.723 G.723.1 G.726 G.728 G.729 G.729.1 G.729a	AC3 ATRAC FLAC iLBC Monkey's Audio Musepack RealAudio SHN Speex Vorbis WavPack WMA ADPCM Nellymoser TwinVQ
Compresión de Imágenes	N/A ¹	JPEG JPEG 2000 JPEG-LS JBIG JBIG2		BMP GIF ILBM PCX PNG TGA TIFF WMP

Tabla 1.-. Resumen de protocolos de codificación (*códec*).

La mayoría de los *archivos de video multimedia* contienen tanto datos de audio como de vídeo, y a menudo alguna referencia que permite la sincronización del audio y el vídeo. Cada uno de estos tres flujos de datos puede ser manejado con programas, procesos, o hardware diferentes; no obstante, para que estos flujos sean útiles para almacenarlos o transmitirlos, deben ser encapsulados juntos. Esta función se realiza mediante la definición de un formato de archivo de vídeo (*contenedor multimedia*) [35]. Por tanto, un concepto importante dentro de la codificación de archivos multimedia es el **contenedor multimedia**. Grosso modo la especificación de un *contenedor multimedia* no son más que unas reglas que empaquetan los diferentes elementos (video, audio, imágenes y, en algunas ocasiones, subtítulos) de forma intercalada (*interleave*) con una determinada compresión (codificación). De esta manera, cada fragmento de archivo tiene suficiente información como para reproducir unos pocos fotogramas, junto con el sonido y subtítulo correspondiente.

¹ No Aplica (N/A)

Algunos de los *contenedores multimedia* más relevantes se enumeran a continuación:

Contenedores Multimedia	Nombre
Audio y Video	3GP
	ASF
	AVI
	Ogg
	Ogg Media
	MP4
	Matroska
	FLV
	MXF
	NUT
	QuickTime
	RealMedia
Sólo Audio	AIFF
	AU
	WAV

Tabla 2.-. Contenedores Multimedia.

Ciertos contenedores son de uso genérico y no limitan el uso de los *códec* o algoritmos en los que se ha codificado el video y el audio. Otros, por el contrario, tienen la limitación de estar orientados a una serie de *códec* y, por tanto, a una compresión de audio y video específicos.

A continuación, se presenta un resumen de los contenedores más representativos, muchos de ellos ampliamente utilizados en el ámbito web:

- **Contenedor AVI** (*Audio Video Interleave*) [36] [37]: Es un formato contenedor de audio y video lanzado por *Microsoft* en 1992. El formato *avi* permite almacenar simultáneamente un flujo de datos de video y varios flujos de audio. El formato concreto de estos flujos (codificación de audio y video) no es objeto de AVI y debe ser interpretado por un programa externo que implemente el *códec*. Es decir, el audio y el video contenidos en el archivo *avi* pueden estar en cualquier formato (*AC3/DivX*, u *MP3/Xvid*, entre otros).
- **Contenedor OGG** [38]: Es un contenedor orientado a *Streaming Media*. Esta orientación a *Streaming Media* es la mayor diferencia en diseño sobre otros formatos contenedores basados en archivo. El flujo de bits de *Ogg* está definido en la *RFC 3533* [39] y el tipo *MIME* recomendado para los archivos *Ogg* es '*application/ogg*' definido en la *RFC 3534* [40]. Oficialmente *Ogg* soporta únicamente *codecs* de vídeo y audio dentro de las especificaciones de la *Fundación Xiph.org* [41] entre los que se encuentran *Theora*, *Vorbis*, *Speex* y *FLAC*.
- **Contenedor OGM** (*Ogg Media*) [42]: Es un contenedor multimedia cuya función es contener el audio (originalmente en formato *Vorbis*), el vídeo (normalmente *DivX*) y subtítulos. *OGM* no es oficialmente soportado por la *Fundación Xiph.org*, debido a que se sale de las especificaciones *Ogg*, así que mientras es ciertamente útil para algunas aplicaciones no es en realidad un contenedor *Ogg*.
- **Contenedor ASF** (*Advanced Streaming Format*, posteriormente renombrado a *Advanced Systems Format*) [43]: Es un formato contenedor de audio y video digital propiedad de *Microsoft*, diseñado especialmente para el *Streaming*. El formato no especifica cómo debe ser codificado el video o audio, en vez de eso sólo especifica la estructura del flujo de video/audio. Lo que esto significa es que los archivos *ASF* pueden codificarse con prácticamente cualquier *códec* de audio/video sin que deje de ser formato *ASF*.
- **Contenedor MP4** (*MPEG-4 Parte 14*) [44]: Es un formato contenedor especificado como parte del estándar internacional *MPEG-4* de *ISO/IEC*. Se utiliza para

almacenar los formatos audiovisuales especificados por *ISO/IEC* y el grupo *MPEG (Moving Picture Experts Group)* al igual que otros contenedores multimedia. Por otra parte, *MPEG-4* es una serie de *codecs* y estándares internacionales de vídeo, audio y datos creado especialmente para la web. Está formado por una serie algoritmos de compresión que codifica datos, audio, y vídeo optimizando su calidad de almacenamiento, codificación y distribución en redes. Con las cámaras de hoy se integra, incluso, captura y codificación en una sola acción, lo que optimiza la potencialidad del usuario para emitir.

Es posible enviar prácticamente cualquier tipo de datos dentro de archivos de este contenedor (*mp4*) por medio de los llamados flujos propietarios, pero los formatos recomendados, por razones de compatibilidad son los especificados por *MPEG-4*: *MPEG-4*, *MPEG-2* y *MPEG-1* para vídeo; *MPEG-4 AAC*, *MP3*, *MP2*, *MPEG-1 Part 3*, *MPEG-2 Part 3* o *TwinVQ* para audio; *JPEG*, *PNG* para imágenes y *MPEG-4 Timed Text* para subtítulos.

- **Contenedor Matroska** [45]: *Matroska* es un formato contenedor estándar abierto que puede contener un número ilimitado de vídeos, audio, imágenes o pistas de subtítulos dentro de un solo archivo. Su intención es la de servir como un formato universal para el almacenamiento de contenidos audiovisuales comunes, como películas o programas de televisión. *Matroska* es similar, en concepto, a otros contenedores, como *AVI*, *MP4* o *ASF*, pero es totalmente abierto. Los archivos de tipo *Matroska* son *.mkv* para vídeo (con subtítulos y audio), *.mka* para archivos solamente de audio y *.mks* sólo para subtítulos.
- **Contenedor FLV (Flash video)** [46] [47]: Es un formato contenedor propietario de *Adobe*, usado para transmitir video por *Internet* usando *Adobe Flash Player*. Los contenedores *FLV* pueden ser incluidos dentro de las *aplicaciones swf* o reproducidos directamente.

A continuación (Tabla 3), se muestran una las versiones de *Flash Player* y algunas de sus características de las codificaciones de video soportadas:

Versión	Año de Desarrollo	Formato de Archivos	Formatos de Codificación de Video	Formatos de Codificación de Audio
6	2002	SWF	Sorenson Spark Screen video	MP3 ADPCM Nellymoser
7	2003	SWF FLV	Sorenson Spark Screen video	MP3 ADPCM Nellymoser
8	2005	SWF FLV	On2 VP6 Sorenson Spark Screen video Screen video 2	MP3 ADPCM Nellymoser
9	2007	SWF FLV	On2 VP6 Sorenson Spark Screen video Screen video 2 H.264	MP3 ADPCM Nellymoser, AAC
		F4V ISO base media file format	H.264	AAC MP3
10	2009	SWF FLV	On2 VP6 Sorenson Spark Screen video Screen video 2 H.264	MP3 ADPCM Nellymoser Speex AAC
		F4V ISO base media file format	H.264	AAC MP3

Tabla 3.- Tipo de codificación soportada por los formatos de archivos que reproduce Adobe Flash Player.

A modo de resumen, las particularidades más importantes:

- Codificación de Video: Los archivos *FLV* contienen un flujo de bits de vídeo que son una variante del estándar *H.263*, bajo el nombre de *Sorenson Spark*. A partir de *Flash Player 8* y las nuevas versiones se soportan la reproducción de vídeo *On2 TrueMotion VP6* y *H.264*. *On2 VP6* puede proveer más alta calidad visual que *Sorenson Spark*, especialmente cuando se usa una *tasa de bits* menor. Por otro lado, es computacionalmente más complejo.
- Codificación de Audio: El audio en los archivos *FLV* se encuentra regularmente codificado como *MP3*. Sin embargo, los archivos *FLV* grabados con el micrófono del usuario usa el códec propietario *Nellymoser*. Los archivos *FLV* también soportan audio sin comprimir o formato de audio *ADPCM*.

Flash Player tiene soporte para un lenguaje de programación llamado *ActionScript*, que puede ser usado para mostrar *Flash Video (FLV)* desde un archivo *swf*. *ActionScript* soporta descarga progresiva mediante *HTTP Streaming* desde la versión de *Flash Player 7*. En la versión de *Flash Player 8*, se permite el acceso aleatorio dentro a los archivos de vídeo utilizando la descarga parcial de *http* y, por ello, se ha denominado, en ocasiones *Streaming*.

Para la reproducción de contenido mediante *Flash Player* se tienen, dos alternativa para la reproducción: *RTMP (Streaming Media)* o *HTTP (Pseudo*

Streaming). Para poder utilizar la reproducción de contenido mediante *RTMP* es necesario la inclusión de nuevos servidores. Sin embargo, a diferencia del *Streaming Media* usando *RTMP*, la reproducción mediante *HTTP Streaming* no soporta transmisiones en tiempo real.

Debido a que el *Flash Player* se ejecuta como un *plugin* del navegador web, es posible incluir *aplicaciones swf* o videos *flv* en páginas web y reproducir los videos por medio del navegador. Esta característica tan importante será útil a la hora de establecer una elección. Por tanto, el *HTTP Streaming* aporta varias ventajas para la reproducción de contenido, entre ellas el uso de *búfer*, utilización de servidores *http* genéricos o la habilidad de utilizar un único *reproductor swf* para reproducir diferentes orígenes y tipos de contenido multimedia (*flv, swf, jpeg, bmp,...*).

Se ha de tener en cuenta que estos contenedores multimedia se establecen para almacenar los archivos de una forma específica; no obstante, no definen el método de transportarlos en una red.

Para el caso de transportar el contenido en redes *IP* existen otros estándares que definen la forma de hacerlo. Por ejemplo, *ISO/IEC 14496-8:2004 (MPEG-4 Parte 8)*, especifica el formato para transportar contenido multimedia por redes *IP* para *MPEG-4*. También incluye las especificaciones para diseñar la carga sobre *RTP* entre otras especificaciones.

Por otra parte, existen varios tipos de contendores en *MPEG-2* los cuales deben ser considerados por su popularidad y uso extendido: *ES (Elementary Stream)*, *PS (Program Stream)* y *TS (Transport Stream)*. Cuando se reproduce, por ejemplo, un video *MPEG-2* de un *DVD*, el flujo está compuesto por varios flujos (llamados *flujos elementales, ES*): uno para el video, uno para el audio, otro para subtítulos, y así sucesivamente. Estos flujos distintos se juntan para formar un único *flujo de programa (PS)*. Así que, los ficheros *VOB (DVD-Video Object o Versioned Object Base)*, que se pueden encontrar en un *DVD*, son realmente ficheros *MPEG-PS* y será, por tanto, un contenedor multimedia tal como se ha definido [48]. Pero este formato *PS* carece de adaptación para la difusión de video a través de una red o por satélite, por ejemplo. Así que, otro formato llamado *Transport Stream (TS)* se diseñó para la difusión de videos *MPEG* a través de estos escenarios [6].

2.7 Resumen

En este capítulo se ha considerado la definición de *Difusión Selectiva* según el concepto anglosajón *Narrowcasting*. En la definición considerada de *Difusión Selectiva* no se establecen criterios o estándares de cómo obtener el servicio, sino que en el contexto actual cada fabricante que lo proporciona lo establece según su criterio. Es por ello que se ha presentado un contexto para el diseño y la elección de la tecnología.

Se ha identificado, dentro del ámbito del *Marketing*, un enorme potencial para estos servicios, por lo que es seguro que será una herramienta que se comenzará a usar de forma progresiva hasta llegar a la generalización (debido, posiblemente, a los intereses económicos que se espera que puedan recaer sobre las empresas que lo utilicen). Por tanto, se intuye que las principales aplicaciones estarán orientadas al *Marketing* a través de lo que se denominan aplicaciones de *Cartelería Digital* y será ésta la aplicación inicial escogida para desarrollar el sistema.

Se han encontrado sistemas de *Difusión Selectiva* contruidos bajo muy distintos tipos de enlaces de comunicaciones. Así, es posible establecer conceptualmente arquitecturas

funcionales de *Difusión Selectiva (Narrowcast)* basadas en enlaces *Unicast*, *Multicast* o *Broadcast*. También, se ha presentado como parte del contexto del proyecto, los tipos de redes (*WAN*, *LAN*) que pueden influir en el diseño y desarrollo de la aplicación y se han definido diferentes tipos de difusión selectiva: *Difusión Selectiva Pasiva*, *Difusión Selectiva Activa* y *Difusión Selectiva Interactiva*, donde el elemento diferenciador es la posibilidad de interacción con el público que se encuentra frente a los medios que presenta el contenido multimedia.

Bajo esta perspectiva se han estudiado las alternativas para transportar el contenido multimedia desde un equipo a otro para su reproducción. Se ha identificado que son varios los criterios a considerar: los *codecs* con los que se han codificado los flujos de datos (principalmente audio y video), los *contenedores multimedia* en los que se almacenan esos flujos en el equipo origen, los *protocolos de red* utilizados para el transporte del contenido multimedia y, finalmente, los métodos *Streaming* para reproducirlos.

Existen varias alternativas a la hora de establecer *Streaming Media*, entre ellas la combinación de *UDP* y *RTP* o *RTSP* que hacen que las entregas de paquetes de datos, desde el servidor al equipo que reproduce el archivo, se hagan con una velocidad mayor que la que se obtiene por *TCP* y *HTTP*. La eficiencia se alcanza por una modalidad que favorece el flujo continuo de paquetes de datos [49]; cuando *HTTP* sufren un error de transmisión, vuelve intentar transmitir los paquetes de datos perdidos hasta conseguir una confirmación de que la información llegó en su totalidad, debido a que el protocolo de transporte es *TCP*. Sin embargo, *UDP* continúa enviando los datos sin tomar en cuenta los fallos en la entrega, ya que en una aplicación multimedia estas pérdidas, si no son excesivas, son prácticamente imperceptibles. Esta eficiencia combinada con la configuración de ciertos protocolos de red para proporcionar *Multicast* [50][51] permite hacer un uso eficiente del *ancho de banda*.

Se ha de tener en cuenta también que el flujo *RTSP Streaming Media* podría fallar, si entre el servidor se encuentra un firewall que bloquee los protocolos y puertos que usa *RTSP*. Estos problemas pueden ser frecuentes en las conexiones a *Internet* mediante una conexión típica en las que se proporciona el *gateway* preconfigurado, incluso si se ha proporcionado *NAT* a *RTSP*, el *Streaming* podría interrumpirse [49]. Estas dificultades en la configuración de los equipos no suelen aparecer cuando la conexión es a través *HTTP Streaming* debido a que usualmente este protocolo viene configurado de forma apropiada en la mayoría de los *gateways* y *firewalls*.

Capítulo 3 Soluciones para Proporcionar Servicios de Difusión Selectiva

En este capítulo se presenta de forma general las soluciones encontradas para proporcionar servicios de *Difusión Selectiva*, así como determinados *arquitecturas funcionales* desarrolladas por algunos fabricantes para obtener escenarios de *Difusión Selectiva* (*Narrowcasting*).

Una vez se hayan definido las *arquitecturas funcionales* identificadas para la distribución del contenido multimedia, se realizará un estudio más exhaustivo sobre uno de los fabricantes, *Cisco*, el cual es siempre un referente tecnológico frente al resto de competidores en conceptos de comunicaciones y redes unificadas. Hay otros desarrolladores importantes de arquitecturas y sistemas que se centran, dentro de la *Difusión Selectiva*, en la *Cartelería Digital*. Estos fabricantes de *software* dejan de lado otros servicios que se pueden ofrecer con, básicamente, las mismas herramientas y *arquitectura funcional*. Por ello, en el estudio de una solución de referencia, nos centramos en una solución de *Cisco* completa y existente en el mercado, la cual abarca la *Difusión Selectiva* como un problema mayor que la *Cartelería Digital*, y nos proporciona una buena referencia para nuestro propio trabajo.

3.1 Introducción a las Soluciones de Difusión Selectiva

Sin duda, en nuestra sociedad, ya ha habido ejemplos prácticos de soluciones de *Cartelería Digital* improvisada incluidas desde hace ya tiempo en nuestro entorno. Yendo al extremo de la simpleza arquitectónica funcional de la solución, posiblemente hayamos visto alguna vez un monitor reproduciendo el contenido de un disco *DVD* en alguna feria o acto ajustando ese contenido promocional a la audiencia potencial.

Dentro de la definición considerada de *Difusión Selectiva*, la solución más simple (para obtener la distribución de contenido remotamente), se podría lograr con una aplicación de reproductor de videos y un acceso remoto al equipo, el cual difundiría contenido multimedia localmente a través de la tarjeta gráfica hacia el medio de visualización. Dicho contenido no podría variarse en tiempo real ni, por supuesto, adaptarse al entorno. La diferenciación del contenido de un equipo a otro, se podría garantizar modificando explícitamente la lista de reproducción de la aplicación que ejerce de reproductor de videos. Por ejemplo, el mecanismo descrito podría lograrse mediante la aplicación *VideoLAN* [52], a través de algún modo de conexión remota como puede ser *VNC* [53] o el *Terminal Server* desarrollado por *Microsoft* para *Windows*. Bajo esta perspectiva tan simple, se pueden establecer variantes que modifiquen las listas de reproducción para la aplicación *VideoLAN* a través de páginas web mediante la interfaz *http* del reproductor *VLC de VideoLAN* [54].

Así, existe en el mercado del *Software* algunas soluciones que intentan automatizar el proceso de distribuir contenido para obtener un sistema de *Difusión Selectiva Pasiva* tal y como hemos definido en la introducción del presente trabajo (ver apdo. 2.5.1 Difusión Selectiva Pasiva).

Actualmente, las empresas desarrolladoras de soluciones de *Difusión Selectiva*, en general, han centrado sus esfuerzos en la implementación de sistemas orientados a la *Cartelería Digital* (*Digital Signage*). La mayoría de las grandes empresas también han desarrollado soluciones de este tipo para proveer a sus clientes de una *arquitectura funcional* adecuada. De este modo pueden establecer una red de difusión y les permite presentar sus productos de forma más espectacular.

Por esta razón, se presenta en este capítulo un resumen de las tecnologías y estrategias de las soluciones que ofrecen algunas empresas y, además, se resume el sistema completo de un referente tecnológico como es *Cisco*, el cual lanza su línea de negocio en tres grandes bloques: *Cartelería Digital*, *Televisión Corporativa* y portal *Portal Web de Vídeo*. Así, el aspecto más interesante que presenta *Cisco*, no es ya su interfaz y las posibilidades que ofrece de configuración, sino cabría destacar el hecho de que implementa una *arquitectura funcional* que une esfuerzos para que la aplicación que hace uso de la arquitectura *consume* el menor *ancho de banda* posible (fundamentalmente *WAN*) cuando el sistema está en pleno rendimiento.

En capítulos posteriores (Capítulo 7 Comparativa: Arquitectura Funcional Implementada y Arquitectura Funcional de Referencia) se realizará una comparación a grandes rasgos entre la solución estudiada y la solución que se ha implementado en este trabajo. Es por ello que se dedica en este capítulo un apartado donde se explica la solución propuesta por este fabricante de *hardware* y *software*.

Ejemplos de Empresas de aplicaciones de Difusión Selectiva y/o Cartelería Digital

A continuación, se proporciona un pequeño listado de ejemplos de pequeñas empresas que proporcionan servicios (*software* y/o *hardware*) relacionados con la *Difusión Selectiva* y/o la *Cartelería Digital*:

<i>Empresa</i>	<i>Web</i>	<i>Servicios</i>
Open Directory		
Admira DN	http://www.admiradigitalsignage.com	Cartelería Digital. Hardware y software.
Anode	http://firesign.net/	Difusión Selectiva. Hardware y software.
DT Research	http://www.dtri.com/	Cartelería Digital. Hardware y software.
Digital Signage EasyStart	http://www.simplifiedigitalsignage.com/	Cartelería Digital. Hardware y software.
EZ-DigiSign	http://www.ezdigisign.com/	Cartelería Digital. Hardware y software.
EnQii Holdings Plc.	http://www.enqii.com/	Cartelería Digital. Hardware y software.
MediaTile	http://www.mediatile.com/	Difusión Selectiva. Hardware y software.
POP AdVisions Inc.	http://www.popadvisions.com/	Cartelería Digital. Hardware y software.
Retail Computing	http://www.radicalcomputing.com/	Difusión Selectiva. Hardware y software.
SVSi, Inc.	http://svsi-volante.com/	Cartelería Digital. Hardware y software.
Sedao Ltd.	http://www.sedao.co.uk/	Difusión Selectiva. Hardware y software.
SportsPageTV	http://www.sportspagetv.com/	Difusión Selectiva. Hardware y software.
TJG digital signage	http://www.tjgtechsales.com/	Cartelería digital. Hardware y software.
Wireless Ronin Technologies Inc.	http://www.wirelessronin.com/	Difusión Selectiva. Hardware y software.
Xstream Solutions	http://www.xstreamsolutions.com/	Difusión Selectiva. Hardware y software.
YouPromote	http://youpromote.com.au/	Cartelería Digital. Hardware y software.
inLighten Inc	http://www.inlighten.net/	Difusión Selectiva. Hardware y software.
Otros		
Fractalia	http://www.fractaliasystems.com/	Difusión Selectiva. Hardware y software.
Exterity	http://www.exterity.com/	Difusión Selectiva e IPTV. Hardware y software.
Adwindow	http://www.adwindow.net/home.html	Cartelería Digital. Hardware y software.
dZine	http://www.dzine.be/	Cartelería Digital. Hardware y Software.
Gimage TV	http://www.gimage.es/	Difusión Selectiva. Hardware y software.
Multipistas	http://www.multipistas.com	Difusión Selectiva. Software.
ShowSpot	http://www.showspot.tv/	Cartelería Digital. Software.
AsomaTV	http://www.asomatv.com	Cartelería Digital. Software.
Soluciones Tecnológicas Integradas	http://www.stidigital.com/	Cartelería digital. Hardware y software.
Comunicads	http://comunicads.com/	Difusión Selectiva. Hardware y software.
DISE	http://www.displayevolution.com/	Cartelería Digital. Software.
Spinetix	http://www.spinetix.com/	Cartelería Digital. Hardware y Software.

Tabla 4.-. Listado de empresas de Cartelería Digital en Open Directory Project y Otras soluciones disponibles.

En la tabla anterior se muestran algunas de las soluciones disponibles con un breve resumen de los servicios que proporcionan; sin embargo, existen otras muchas.

3.2 Características de las Soluciones Encontradas

Como se ha comentado ya, existen varios tipos de soluciones por lo general divergentes y propietarias a cada uno de los fabricantes. La causa de esta gran variedad es que existen diferentes formas de implementar una solución de *Difusión Selectiva*, limitada cada una de ellas por la tecnología utilizada o la imaginación del equipo desarrollador.

Según se escoja una u otra solución se tiene una serie de ventajas e inconvenientes que hay que valorar, no existiendo la solución perfecta ya que ésta también se debe amoldar a la aplicación o problema concreto y específico al que se quiere encontrar solución. Por una parte, un pequeño comerciante no debería estar interesado en un desarrollo basado en potentes servidores y una arquitectura compleja y, por otra parte, una gran corporación no se debería centrar en una solución poco evolucionada que le haga consumir muchos recursos de red. Es decir, una organización con varias sedes distribuidas geográficamente ponderará con un coeficiente alto a la solución que consuma el *ancho de banda* entre sedes de forma eficiente; así, es una cualidad importante el disponer del mayor *ancho de banda* contratado con los operadores, el cual es un bien escaso y finito, para utilizarlo en otras aplicaciones como son las *Comunicaciones Unificadas* (ToIP, Video Conferencias, presencia, Tele-presencia, acceso a aplicaciones, acceso a Internet,...).

Por tanto, antes de tomar una decisión del sistema a implantar dentro de la organización se ha de plantear una serie de cuestiones básicas antes de que el Ingeniero se decida por una u otra alternativa:

- ¿Cuánto *ancho de banda* consumen las diferentes soluciones?
- ¿Se tendrá que ampliar el *ancho de banda* disponible entre sedes? ¿En qué costes repercutirá?
- ¿Es necesario un sistema centralizado o distribuido?
- ¿Cuántas ubicaciones geográficas diferentes mostrarán contenido?
- ¿Es necesaria *Difusión Selectiva* o todos los escenarios mostrarán el mismo contenido?
- ¿Es la solución escalable? ¿Se necesita una solución escalable?
- ¿Es necesario implementar *Calidad de Servicio*? ¿Convive la solución con tráfico de red prioritario?
- ¿Son necesarios requisitos de difusión de contenido en tiempo real?
- ¿Basta con una solución de *Cartelería Digital*?
- ¿Es necesaria interactividad entre público objetivo y el escenario?
- ¿Sería interesante/viable replantear la arquitectura de red de la organización?

Éstas y muchas otras cuestiones son las que se debería plantear la dirección de la organización para escoger la solución más adecuada y, así, dependiendo de las respuestas, serían unas u otras soluciones la más conveniente.

Un ejemplo significativo, en la que se refleja la importancia de la solución adoptada, es la red establecida en *Metro de Madrid*. En ella se emite el mismo contenido en cada una de las pantallas presentes en los diferentes andenes de las estaciones que lo forman. Es un sistema *Broadcast o Multicast*, cuya información llega en diferentes instantes de tiempo². Con esta solución un cliente de la *Red de Metro de Madrid* podría anunciarse en cada una de las pantallas desplegadas por el sistema llegando a un gran número de público de forma casi simultánea. Desde el punto de vista de la Ingeniería, o más concretamente el mío propio,

² Observación física realizada en diferentes emplazamientos simultáneamente sobre la red de *Metro de Madrid* a finales del año 2009.

hubiese sido mucho más interesante establecer un sistema *Narrowcast*, pero antes, el ingeniero debería haberse planteado algunas preguntas (*¿Es necesaria Difusión Selectiva o todos los escenarios mostrarán el mismo contenido? ¿Son necesarios requisitos de difusión de contenido en tiempo real?*). En parte es debido a que una solución *Narrowcast* también permite emisión similar a la *Broadcast* o *Multicast* aunque, dependiendo de la solución, puede estar algo desfasada o desincronizada de unas estaciones clientes a otras, como ocurre actualmente en la arquitectura implantada en la *Red de Metro de Madrid*.

Centrándonos en el ejemplo considerado, con un sistema *Narrowcast*, podría emitirse diferente contenido en cada línea de metro específica. Es más, además de esa diferenciación por línea, podría emitirse contenido diferente de forma geográfica de modo que podrían establecerse regiones, o como se ha definido en este trabajo y como veremos en la terminología de este documento: *Ubicaciones*. Así, un anunciante podría estar interesado sólo aparecer en ciertas estaciones debido a la proximidad de su negocio y otro solamente en los barrios de la periferia. El abanico de posibilidades para *Metro de Madrid* se abriría de forma significativa ofreciendo un servicio diferenciado, abriéndose a nuevos servicios de facturación e introduciéndose de forma relativa en la ventaja de la *Economía de Colas Largas*³ (*"The Long Tail"*).

También, inevitablemente, es conveniente tener al menos aprovisionada la posibilidad de emitir contenido en directo con el objetivo de obtener las funciones de informar al público en casos de emergencias o de información de interés de forma instantánea. Es por todas estas ideas que se considera que la elección del sistema puede ser un elemento muy importante dentro de la organización.

En los sucesivos sub apartados, se presentan algunos tipos de soluciones encontradas, remarcando las características generales y más comunes a las diferentes implementaciones de cada *arquitectura funcional* revisada.

3.2.1 Aplicaciones Software de Propósito Específico y aplicaciones de Carácter General

Existen diferentes aplicaciones, por un lado están las más comerciales, las cuales se desarrollan con carácter general para un número amplio de servicios y del otro las que han sido implementadas para satisfacer una necesidad concreta; en esta última, suelen ser aplicaciones por encargo que necesitan algún tipo de integración con otro *software* o *hardware*.

Dentro de las *Aplicaciones de Propósito Específico* se suelen incluir aplicaciones de *Difusión Selectiva Interactiva* y *Aplicaciones de Difusión Selectiva Activas* mientras que las de *Carácter General* no suelen implementar sistemas de interacción con el público, en general, y se centran en la *Difusión Selectiva Pasiva*, tal y como se han definido.

³ A modo de resumen indicar que una *Economía de Colas Largas* implica que puede haber muchos clientes interesados en anunciarse en pocas estaciones (debido a su proximidad geográfica con la estación y aplicada al caso concreto del ejemplo). Así, al haber muchos clientes, la facturación puede ser mayor que anunciar a pocos clientes de forma global. Esta nueva forma de entender el mercado potencial es uno de los mayores baluartes de la venta de comercio electrónico por Internet [158].

A continuación se muestra una tabla con las características generales más significativas de las soluciones encontradas de *Propósito Específico* y de *Carácter General*:

Características Generales	
Aplicaciones de Propósito Específico	Aplicaciones de Carácter General
<ul style="list-style-type: none"> • Desarrolladas por PYMES. • Difusión Selectiva Interactiva y Activa. • Integración específica con otros sistemas propietarios ya sean <i>Hardware</i> o <i>Software</i>. 	<ul style="list-style-type: none"> • Desarrolladas por fabricantes y grandes corporaciones⁴. • Difusión Selectiva Pasiva. • Integración estandarizada con otros sistemas relacionados con la Difusión Selectiva (por ejemplo, Servidores de Streaming, Directorio Corporativo LDAP,..)

Tabla 5.-. Características Generales de las Aplicaciones de Difusión Selectiva atendiendo al Propósito Específico o al Carácter General

3.2.2 Arquitecturas Cliente-Servidor y Arquitecturas Distribuidas

Se puede hacer una diferenciación de arquitecturas en función del rol, dentro de la arquitectura desarrollada, en la lógica de funcionamiento de los diferentes participantes en la comunicación. Es posible que la arquitectura se encuentre desarrollada bajo una filosofía *cliente-servidor*, de modo que el contenido se muestre en función de las peticiones de los *equipos terminales* que serán los equipos o sistemas que finalmente muestren visualmente el contenido. La otra alternativa es que el contenido se distribuya a través de todos los miembros pertenecientes a la red. La distribución se puede realizar a través de sistema *P2P*, de modo que todos los nodos que forma la *Red de Difusión Selectiva* presentan el mismo rol y tienen el mismo sistema de funcionamiento. Estos nodos se comunican entre sí adoptando el contenido de forma diferente en función de la configuración de cada equipo. En este último tipo de sistemas, prima la eficiencia en el envío y distribución de contenido multimedia, intentando hacer un uso eficiente del *ancho de banda* disponible. Existe una gran variedad de alternativas para lograr esta eficiencia.

A continuación (Tabla 6), se presentan las características más destacables:

Características Generales	
Arquitectura Cliente-Servidor	Arquitectura Distribuida
<ul style="list-style-type: none"> • Configuración centraliza de todos los clientes. • Existencia de un equipo con una alta carga de procesamiento. • Facilidad en el control y detección de fallos en el sistema. 	<ul style="list-style-type: none"> • Utilización del ancho de banda disponible de forma eficiente. • Configuración específica a cada elemento. • Dificultad en el control y en el mantenimiento del sistema.

Tabla 6.-. Características Generales de las Arquitecturas atendiendo a si establecen un Sistema Cliente-Servidor basado en peticiones o una Arquitectura Distribuida.

Dentro de la *Arquitectura Cliente-Servidor* se encuentra una ventaja importante. Esta ventaja es el hecho de tener un sistema centralizado, ya que aunque un único punto de fallo hace depender a todo el sistema del funcionamiento de un solo elemento, una planificación apropiada de *Backup (Copia de Seguridad)* y replicación de información, hace que el sistema sea muy robusto ante fallos del sistema. Además, dentro de este tipo de sistemas basados en una *Arquitectura Cliente-Servidor* se pueden implementar sistemas de alta disponibilidad con alta redundancia. Por otra parte, nos encontramos que las *Arquitecturas Distribuidas* muestran además de una mayor complejidad, una mejora en la utilización de recursos ya que reparten la carga entre cada uno de los miembros que componen la arquitectura. En próximos apartados se muestra, conceptualmente además de la inclusión de algunas técnicas en el desarrollo del

⁴ Se ha detectado a la conclusión del proyecto una pequeña tendencia de algunos grandes fabricantes que está incluyendo elementos de interactividad.

Marco implementado, como aplicando otras técnicas a una *Arquitectura Cliente-Servidor* es posible mejorar el aprovechamiento del *ancho de banda* disponible en el sistema manteniendo de forma centralizada la configuración de los equipos.

También comentar que es posible desarrollar un híbrido de estas dos líneas de trabajo, aunque un desarrollo de un sistema así puede quedar mejor enmarcado dentro de otros tipos de arquitecturas como las que siguen.

3.2.3 *Arquitecturas Auto-Proxy o Bajo Demanda*

La diferenciación que se presenta en este apartado es clave, ya que según sea el sistema implementado, lograr una *Arquitectura Auto-Proxy* establece una clara mejora en el uso de recursos de red: fundamentalmente mejoras muy importantes en el aprovechamiento del *ancho de banda* que, como hemos indicado, es un bien escaso.

Una *Arquitectura Auto-Proxy*, tal como se define en este documento, mantendrá almacenada de forma local (en el *equipo terminal*) el contenido a emitir que previamente se ha descargado del *equipo servidor de contenido multimedia*. Así, en la *Arquitectura Bajo Demanda*, el *equipo terminal* le pide al *equipo servidor*, el contenido a emitir cada vez que éste aparece en la programación; mientras en las arquitecturas que se han denominado *Auto-Proxy*, solamente se descarga el contenido una vez, aunque éste esté programado de forma repetida.

La implementación de estos dos tipos de arquitecturas queda incluida dentro del paradigma de *Arquitectura Cliente-Servidor* y son implementadas en fases maduras del desarrollo de la arquitectura: cuando se consolida como un sistema utilizable y/o comercializable.

Para ilustrar como puede mejorar, una arquitectura *Auto-Proxy*, el uso eficiente del *ancho de banda* se va a recurrir a un ejemplo.

Imaginemos pues, que estamos ante un sistema de *Difusión Selectiva* con arquitectura inicial *Cliente-Servidor* en un escenario de *Difusión Selectiva Activa* (ver apdo. 2.5.2 *Difusión Selectiva Adaptativa o Activa*) en un marco de *Cartelería Digital* (uso más extendido de la *Difusión Selectiva* y aplicación inicial por excelencia). El contenido que se emite en una pantalla o monitor o, incluso, en un escaparte a través de un proyector, depende de un sistema basado en cámaras infrarrojas. Este sistema es capaz de discernir, a través de los rasgos faciales de la persona que se presenta delante del medio de difusión, si esa persona es de sexo masculino o femenino. A partir del mecanismo de decisión (basado por ejemplo en métodos estadísticos y de entrenamiento) el sistema conectado a la cámara infrarroja, cuando detecta que la persona (o la mayoría de personas) que se encuentra frente a la pantalla de presentación de contenido es de sexo masculino, emite un vídeo de deportes y si la persona (o la mayoría de las personas) es de sexo femenino decide (por decisión de un hipotético departamento de *Marketing*) que el contenido a presentar debe ser un video relacionado con la moda femenina.

En la situación, por ejemplo, en el que el video relacionado con la moda ya se haya emitido una vez, resulta poco eficiente volver a descargarlo del servidor como hace una *Arquitectura Bajo Demanda*. En el caso de implementar una *Arquitectura de Auto-Proxy*, el video queda descargado en la máquina cliente la primera vez y se mantiene almacenado para futuras programaciones o visualizaciones. De esta forma, cada vez que se presente una persona de sexo femenino delante del sensor infrarrojo no es necesario volver a realizar una petición al servidor.

Como es evidente, puede haber variantes diferentes de estos tipos de arquitecturas, a la que se han denominado en el *proyecto fin de carrera* como *Auto-Proxy* y *Bajo Demanda*, e incluso una combinación de ellas, siendo ésta, elección del equipo de ingenieros de la solución.

A continuación, se presenta una tabla resumen con las características encontradas más significativas:

<i>Características Generales</i>	
<i>Arquitectura Auto-Proxy</i>	<i>Arquitectura Bajo Demanda</i>
<ul style="list-style-type: none"> • Uso eficiente del Ancho de Banda. • Mayor complejidad de la implementación de este tipo de soluciones. • Funcionalidad de soluciones maduras. 	<ul style="list-style-type: none"> • Uso ineficiente del ancho de banda. • Mayor simplicidad en la implementación de la solución. • Funcionalidad de Soluciones iniciales.

Tabla 7.-. Características Generales de las Arquitecturas atendiendo a si establecen un Auto-Proxy o basado en peticiones de contenido Bajo Demanda

Algunas de las soluciones que se han consultado en este trabajo son soluciones *Auto-Proxy* que nacieron de la evolución de una *Arquitectura Bajo Demanda* en sus inicios. Por otra parte, indicar que puede haber un híbrido en la *Arquitectura Auto-Proxy* en el sentido de que tan sólo uno de los *equipos terminales* funcione como *Auto-Proxy* (en esta forma de operar es un auténtico *proxy* de contenido) y el resto de equipos presentes en la red realicen las peticiones a este equipo (o mecanismo de portal cautivo) en vez de realizar las peticiones directamente al servidor.

3.2.4 Arquitecturas Basadas en Software de Aplicación y Arquitecturas Basado en herramientas Web

Es interesante observar que algunas aplicaciones han ido evolucionando de forma que fueron concebidas como aplicaciones desarrolladas para un *sistema operativo* y que posteriormente han ido migrando hasta convertirse en aplicaciones controladas mediante navegadores web. Dentro de esta evolución, nos encontramos aplicaciones de todo tipo, desde herramientas para *virtualizar* equipos [55], hasta aplicaciones para compartir archivos de forma distribuida [56][57] e incluso reproducción de contenido [52].

Por una parte, generar *Arquitecturas Basadas en Software de Aplicación* tiene algunas ventajas, como es la implementación de sistemas seguros en los que se puede definir la forma de comunicación entre la aplicación y el servidor. Sin embargo, para este tipo de arquitecturas hay que desarrollar varias aplicaciones: la aplicación que **controla** la configuración de los *equipos terminales*, la aplicación que **realiza** la configuración del los *equipos terminales* y la aplicación que **muestra** el contenido desde los *equipos terminales*.

Desarrollando una *Arquitectura Basada en Aplicaciones Web* se presentan una serie de dificultades a la hora del desarrollo, principalmente en la integración de las distintas tecnologías utilizadas, en especial la integración de sensores y otros dispositivos. Aún así, el estado actual de las distintas tecnologías permite una integración apropiada sin excesivas dificultades. Por otra parte, se añaden una serie de ventajas como es el acceso a la configuración del servidor de forma remota a través de un interfaz de navegación web. De este modo es innecesario instalar ninguna aplicación especial, por los requisitos de *sistema operativo* son mínimos si éste llevara un navegador instalado; situación que ocurre en la mayoría de los *Sistemas Operativos* de ordenadores de sobremesa convencionales. Además, como característica adicional de interés, el hecho de realizar la implementación del *equipo servidor* y el *equipo terminal* con tecnologías web, permite la migración a los diferentes sistemas operativos sin más que copiar las carpetas y realizar algunos ajustes para el correcto

funcionamiento. Sírvase como ejemplo el desarrollo del presente proyecto, el cual se ha realizado en un entorno de equipos de desarrollo de *Microsoft Windows* a partir de aplicaciones (*Adobe Flash Professional CS3*) que no se ejecutan bajo *sistemas operativos Linux*. Y por otro lado, el sistema final descansa sobre máquinas *Linux*: tanto el *Equipo Servidor de Contenido y Configuración (ESCC)* como el *Equipo Reproductor Cliente (ERC)* o *equipo terminal*.

3.2.5 Arquitecturas con Hardware Genérico, Hardware Específico y Hardware Integrado o Embebido

Independientemente del tipo de *software* desarrollado para establecer la arquitectura, se establece una relación entre el *software* y *hardware* utilizado.

Si se desarrolla un *Hardware Específico* para obtener una *Arquitectura de Difusión Selectiva* determinada, se ha de tener en cuenta que quizás no todas las aplicaciones se pueden ejecutar en esa arquitectura *hardware*. Por ello, es posible que los *sistemas operativos* y las aplicaciones disponibles no puedan ser ejecutados; por lo que muchas veces es necesario, adicionalmente, el desarrollo de un *sistema operativo* y de aplicaciones que puedan ser ejecutadas en el *Hardware Específico* e, incluso, puede ser necesario el desarrollo de un *firmware*.

La utilización de un *Hardware Genérico* (por ejemplo, *Arquitecturas x86*) permite hacer uso de *sistemas operativos* comúnmente utilizados y aplicaciones de uso común, facilitando así las tareas de desarrollo.

Es típico también el desarrollo de aplicaciones en sistemas con *Hardware Integrado o Embebido*, de modo que se instalan *sistemas operativos* en arquitecturas simplificadas de equipos informáticos. Por tanto, un *sistema integrado o embebido* es un sistema informático de uso específico construido dentro de un dispositivo mayor. Los *sistemas integrados* se utilizan para usos muy diferentes de los usos generales para los que se emplea un ordenador personal. En un sistema integrado la mayoría de los componentes se encuentran incluidos en la placa base: la tarjeta de vídeo, tarjeta de audio, tarjeta de red, módem,... Dos de las cualidades principales son el precio y el consumo energético.

Además, los sistemas integrados suelen usar un procesador relativamente pequeño y una memoria pequeña para reducir los costes. Estas ventajas se enfrentan, sobre todo, al problema de que un fallo en un elemento implica la necesidad de reparar la placa integrada.

A continuación, se muestra una tabla resumen que muestra las características generales:

Características Generales		
Hardware Genérico	Hardware Específico	Hardware Integrado
<ul style="list-style-type: none"> • Equipos de alta capacidad de Procesado. • Mantenimiento por piezas. • Aplicaciones y sistemas operativos genéricos. • Facilidad de Desarrollo. • Alto consumo energético. • Tamaño grande. • Integración con sensores amplia. 	<ul style="list-style-type: none"> • Equipos diseñados con la capacidad de procesamiento necesaria al sistema a desarrollar: normalmente pequeña. • Mantenimiento por unidades. • Aplicaciones y sistemas operativos específicos. • Complejidad de desarrollo. • Bajo consumo energético. • Tamaño reducido. • Bajos costes. • Integración con sensores implica rediseño del hardware. 	<ul style="list-style-type: none"> • Equipos con capacidad de procesamiento media. • Mantenimiento limitado por piezas. • Aplicaciones y sistemas operativos Genéricos. • Facilidad de Desarrollo. • Bajo-Medio consumo energético. • Tamaño pequeño. • Medios-Bajos costes. • Integración con sensores media.

Tabla 8.-. Características Generales de las arquitecturas atendiendo al tipo de *hardware* utilizado para los equipos del sistema final.

3.3 Evolución Detectada de las Arquitecturas de Difusión Selectiva

Todas estas arquitecturas definidas, y brevemente descritas como posibles soluciones finales, son algunas alternativas que los ingenieros desarrolladores pueden implementar. Quedan otras no comentadas, pero se ha decidido que éstas son las más importantes.

Así, lo habitual será evolucionar cada solución en una mejora de prestaciones dependiendo de las características del *hardware* disponible y el propósito de la solución; no obstante, esa evolución contendrá una integración de la combinación de las diferentes arquitecturas expuestas, siendo la evolución más usual, según la observación de algunos fabricantes y desarrolladores de *software*, la que se muestra a continuación:

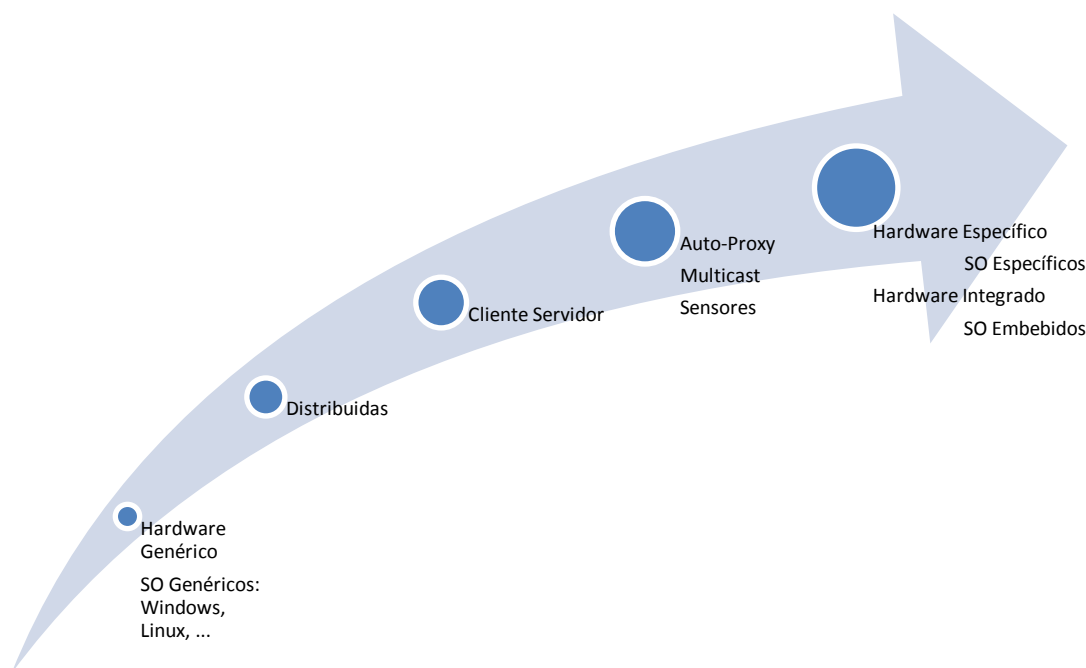


Ilustración 10.-. Evolución esperada en el desarrollo de una solución para aplicaciones de Difusión Selectiva.

Inicialmente los desarrolladores se basan en arquitecturas convencionales con *sistemas operativos* convencionales; donde proveen, normalmente, al ordenador personal de las funcionalidades para programar videos e imágenes.

El segundo paso, considerado como natural, es poder controlar el contenido de forma remota, de modo que se crean aplicaciones que se instalan en las maquinas reproductoras clientes y en los equipos que realizan la configuración de las máquinas, para así poder obtener un control remoto de esas programaciones. En esta evolución, algunos fabricantes de *software* observan que, aunque las configuraciones son independientes, parte del contenido multimedia programado coincide en los diferentes equipos clientes y es compartido entre las máquinas que forman el sistema. Es aparentemente por esta razón por la cual se evoluciona, en algunos casos, al desarrollo de una *Arquitectura Distribuida*.

Una vez que se tiene la posibilidad de un control remoto a través de red, se espera obtener la funcionalidad de programar el contenido de la reproducción a partir de un solo punto de modo que, de esta forma, se tenga acceso y control de todos los elementos de la red, obteniendo una *Arquitectura Cliente-Servidor*. Mediante este tipo de arquitecturas hay que realizar una sola configuración de un equipo y no hay que realizar una configuración de forma distribuida de todos los *equipos terminales* en lo relativo a la programación.

Posteriormente, en la mejora del *software* desarrollado se suele desear utilizar los recursos de la forma más eficiente posible, tanto los recursos de procesamiento, como los de red. Por ello, además de añadir nuevas funcionalidades al *software*, se espera un mayor rendimiento desarrollando *Arquitecturas Proxy* y *Auto-Proxy*. Dentro de estas soluciones, algunos fabricantes inician la inclusión de sistemas *Multicast* y *sensores*. Típicamente las empresas que han realizado estos desarrollos son medianas empresas cuyas aplicaciones están muy orientadas a las necesidades particulares de los clientes.

Por último, los grandes fabricantes de soluciones de *Difusión Selectiva*, en un afán (se intuye) de ahorro de costes y eficiencia en la fabricación de dispositivos, además de realizar una búsqueda de eficiencia en el consumo de energía se inclinan y apuestan por el desarrollo de *hardware* específico. En este punto del desarrollo de la solución, la mayoría de fabricantes optan por añadir, al diseño del *hardware* específico, elementos adicionales para la obtención de nuevas funcionalidades que le permitan desarrollar y ofrecer *Arquitecturas de Difusión Selectiva Activas o Interactivas*. De esta forma se añade interactividad a través *hardware*, como *bluetooth* o insertando cámaras y sensores, para adaptar la programación al entorno donde se encuentren ubicados los dispositivos físicos de emisión de contenido (pantallas, monitores, proyectores, video-wall,...). Un caso muy típico es la introducción de un mando a distancia mediante infrarrojos, elementos táctiles o cámaras que obtengan determinada información (número de personas frente al dispositivo o sexo de *público real*).

3.4 Soluciones de Difusión Selectiva

Hasta ahora muchas organizaciones han tendido a adoptar pequeñas soluciones de múltiples fabricantes: por una parte, se ha utilizado programas de *software* para generar los contenidos digitales y las presentaciones, se han utilizado otros elementos para la reproducción del contenido y también diferente equipamiento para la distribución de este contenido hacia los equipos finales. Esto se ha traducido en integraciones complejas de alto coste en propiedad y limitada capacidad de ampliación. El reto de soportar múltiples formatos, diferentes medios de accesos a la información y métodos de acceso, ha intensificado aún más los planteamientos de toma de decisiones en los *Departamentos de Comunicaciones* de las

organizaciones. Además, muchos de los productos disponibles hoy, limitan el acceso a la información por el desarrollo de *codecs* cerrados especiales y reproductores propietarios.

Esta situación ha creado aparentemente la necesidad de desarrollar una solución integrada que permita abordar toda la cadena de valor de los medios digitales; es decir, crear un *Marco de Aplicaciones de Difusión Selectiva de Contenido Multimedia* que abarque todos estos aspectos: desde una determinada *arquitectura funcional*, hasta las aplicaciones (o la base de las aplicaciones) que utilicen de forma apropiada esa *arquitectura funcional implementada*.

Tal y como se ha expuesto vamos a realizar un análisis completo de una *Arquitectura de Referencia*, la *Digital Media System (DMS)*⁵ de Cisco. La razón es poder comparar las capacidades de la arquitectura desarrollada con las existentes en el mercado, ver también las tendencias de éste y determinar si la *arquitectura funcional* finalmente adoptada y, por tanto, el *Marco de Aplicaciones de Difusión Selectiva* diseñado y desarrollado, es una solución admisible. Así, la *Arquitectura de Referencia* presenta una solución para proveer las funcionalidades (aplicaciones) que se presentan en este proyecto (*Cartelería Digital*) y otras añadidas como se expondrá (*Portal Web de Video* y *Canal de TV Corporativa*), cuyo desarrollo se podrían incluir como continuación de este proyecto como líneas de trabajo futuras o mejoras.

Antes de aventurarnos a implementar las nuevas funcionalidades como líneas de trabajo futuras o mejoras, es conveniente tener bien definidas estas funcionalidades y ver si tienen hueco dentro de los objetivos marcados en este proyecto.

3.4.1 Solución Propietaria de Referencia: Cisco Digital Media System (Suite)

La aplicación escogida como de referencia ha sido la desarrollada por la empresa Cisco, el cual es un fabricante que ha aportado diversos estándares '*de facto*' y '*de forma*' a la industria de las telecomunicaciones. El sistema de *Difusión Selectiva* de Cisco (*Cisco Digital Media System, DMS*) es un conjunto de aplicaciones y dispositivos que de forma conjunta proporcionan *Cartelería Digital*, un *Portal Web de Video* y *Canal de TV Corporativa* [58].

El *Digital Media System*, a partir de ahora **DMS**, ofrece todas las funcionalidades, como veremos, desarrolladas en este proyecto final de carrera. Sin embargo, la complejidad de la arquitectura también se hace evidente como iremos mostrando, siendo esto último un importante escollo para ciertas aplicaciones y, por tanto, no puede considerarse siempre como la elección más adecuada.

A continuación (Ilustración 11), se resumen cada uno de los servicios provistos por la arquitectura de referencia de *Difusión Selectiva* y los elementos más significativos que intervienen:

⁵ En las últimas definiciones de la tecnología se ha rebautizado como *Digital Media Suite*, variando el nombre también de algunas de las aplicaciones; sin embargo, aún no se ha actualizado toda la documentación de la web de Cisco (<http://www.cisco.com>). Fecha diciembre 2010.



Ilustración 11.- Servicios Proporcionados por el sistema de Difusión Selectiva de Cisco y elementos más importantes dentro de la arquitectura funcional. Fuente Cisco 2008-2010.

Como se observa en la Ilustración 11, existe diferente equipamiento para establecer la arquitectura [59] (*Cisco Digital Media Encoder*, *Scientific Atlanta Encoders* y *Digital Media Manager*), así como los dispositivos y elementos (*Video Portal for Desktop Video*, *Digital Media Player for Enterprise TV* y *Digital Media Player for Digital Signage*) para ofrecer los tres diferentes servicios de *Difusión Selectiva*.

Se definen a continuación los elementos presentes en la solución [58]:

Cisco Digital Media Manager (DMM)

Administrador de *Medios Digitales (Contenido Multimedia)*. Es una aplicación de *software* desarrollada como aplicación web que controla la creación, administración y publicación de contenidos multimedia para pantallas en *Cartelería Digital* y otras aplicaciones. El *Digital Media Manager*, a partir de ahora **DMM**, muestra y administra de manera remota grupos y unidades conectadas con el *Digital Media Player* de Cisco, el cual es una especie de *Set-top-Box (STB)*⁶ basado en tecnologías *IP*. El *software* también maneja listas de temas musicales, agendas de contenidos, plantillas personalizadas con zonas en pantalla, control de acceso basado en roles, además de permitir la monitorización y presentación de informes. Añadiendo la posibilidad de incluir un módulo denominado *Portal de Video (Video Portal)*, que ofrece una aplicación para presentar *Videos en los Escritorios (Desktop Video)* de los usuarios registrados.

Cisco Digital Media Player (DMP)

Es un dispositivo de pequeño tamaño y a veces denominado en algunos contextos como *Set-top-Box*, que controla los vídeos, gráficos, audios y textos en los dispositivos que muestran el contenido de *Cartelería Digital* y/o *Canal de TV Corporativo*. En el caso del fabricante de referencia el reproductor se denomina *Digital Media Player*, a partir de ahora **DMP**, y soporta video de pantalla completa o gráficos por zonas, ofreciendo almacenamiento local para una

⁶ **Set-top Box (STB)**, cuya traducción literal al castellano es “aparato que se coloca encima del televisor”, es el nombre con el que se conoce el dispositivo encargado de la recepción y, opcionalmente, decodificación de señal de televisión analógica o digital, para luego ser mostrada en un dispositivo de televisión. Es por ello que este término es utilizado indistintamente para referirse a los elementos que proporcionan el contenido multimedia al medio de presentación en el contexto de la *Difusión Selectiva (Narrowcasting)*.

redundancia automática y de alta disponibilidad: algo similar a lo que se ha definido como *Arquitectura Auto-Proxy* (ver apdo. 3.2.3 Arquitecturas Auto-Proxy o Bajo Demanda).

El sistema *DMP* lleva embebido un *sistema operativo* y estos dispositivos van, de forma general, ubicados dentro de una red siendo miembros de una subred. Pueden trabajar de forma aislada o en grupo, formando parte de una arquitectura más o menos compleja, como veremos (apdo. 3.4.3 Arquitectura Funcional de Referencia).

Cada *DMP* lleva un sistema que permite decodificar el audio y el video, y el escenario es personalizable mediante *zonas*. El *DMP* soporta *MPEG 1, 2 y 4*, tanto en definición estándar, como de alta calidad y pueden ser conectados directamente a *Internet* a través de modem o cable o configurando la interfaz de red LAN, mediante *DHCP* o asignándole una *dirección IPv4 estática*.

Existen tres⁷ modelos de *DMPs*: dos (*DMP 4400G, DMP 4310G*) para proporcionar *Sistema de TV Corporativo (Enterprise TV)* los cuales llevan un mando a distancia y otro (*DMP 4305G*), más económico y con menos funcionalidades, diseñado para proporcionar el servicio de *Cartelería Digital (Digital Signage)* dentro de la *Difusión Selectiva*.

Para desarrollos de amplia escala, los cuales se consideran cuando existen más de 5 *DMPs* dentro del sistema, se debe utilizar un *DMM* para proporcionar una provisión centralizada de todos los *DMPs* ya que se considera que la configuración manual de todos los equipos se hace incómoda. Esto facilita la configuración y provee bastante flexibilidad, pasando el sistema de una *Arquitectura Distribuida* a una *Arquitectura Cliente-Servidor* como se han definido en el apartado 3.2.2.

Digital Media Encoder (DME) y Scientific Atlanta Encoder

Son Codificadores de Señales Digitales y Analógicas (audio y video) que permiten transmitir la información por redes IP, estableciendo la compresión en diferentes *codecs* y/o contenedores multimedia (Windows Media 9, MPEG 4/H.264 Flash: '.flv'—bajo demanda-).

A continuación, se muestran algunas características de los tipos de codificadores que se ofrecen y han sido certificados en la solución de referencia:

Modelo	Configuración máxima recomendada por Cisco
Cisco Digital Media Encoders 2000 & 2200	2 Live stream outputs with a VoD saved per output stream Window Size: 640 x 480 Bit Rate: 2 Mbps
Cisco Digital Media Encoder 1000	Single live stream output with single VoD Window Size: 320 x 240 Bit Rate: 350 Kbps
Cisco Digital Media Encoder 1100	Single live stream output with single VoD Window Size: 640 x 480 Bit Rate: 2 Mbps

Tabla 9.-. Configuraciones recomendadas según codificadores disponibles en la solución de referencia.
Fuente Cisco 2008-2010.

Estos equipos suelen tener integrados *DSPs (Digital Signal Processor)* con funcionalidades concretas para la codificación y decodificación de vídeo, de modo que estos equipos podrían considerarse como de *Hardware Específico* según se ha definido (apdo. 3.2.5 Arquitecturas con Hardware Genérico, Hardware Específico y Hardware Integrado o Embebido).

⁷ Aunque a principios de 2010, aún no habían llegado a España eran modelos que se estaban comercializando ampliamente ya en otros países como EEUU y Canadá.

Una comparativa de los diferentes **DMEs** nos da una idea de las características de estos equipos:

Modelos	Dimensiones (H x W x D)	Entradas Video	Entradas Audio	HD [GB]	Procesadores
Cisco Digital Media Encoder 2200	2 unidades rack (RUs) 22.05 kg (8.89x48.26x58.42cm)	2 composite 2 S-Video 2 SDI	2 balanced stereo (XLR) 2 unbalanced stereo(RCA) 2 digital audio (AES/EBU)	100	Dual AMD Opteron Quad Core, 2.2GHz
Cisco Digital Media Encoder 2000	2 unidades rack (RUs) 19.05 kg (8.89x48.26x60.96 cm)	2 composite 2 S-Video 2 SDI	2 balanced stereo (XLR) 2 unbalanced stereo(RCA) 2 digital audio (AES/EBU)	85	AMD Dual Core
Cisco Digital Media Encoder 1100	Portable 3.41 kg 11.43x19.67x30.48in	1 composite 1 S-Video	1 balanced stereo (XLR) 1 unbalanced stereo(RCA)	100	2.119GHz Intel Core 2 Duo CPU
Cisco Digital Media Encoder 1000	Portable 5.44 kg (10.16x19.67x39.37cm)	1 composite 1 S-Video	1 balanced stereo (XLR) 1 unbalanced stereo(RCA)	40	Intel Celeron

Tabla 10.-. Resumen de características hardware de los diferentes tipos de codificadores de video proporcionados por el fabricante de referencia. Fuente: Cisco 2008-2010.

Se observa que algunos de los modelos, debido a sus dimensiones, deben situarse en un *Data Center* o *CPD* (*Centro de Proceso de Datos*) y son equipos dimensionados para estos emplazamientos. Sin embargo, también se observa que existen modelos para soluciones más sencillas como puede ser el *Cisco Media Encoder 1100* con unas dimensiones mucho más reducidas.

3.4.2 Aplicaciones de la Solución de Referencia

Como se ha adelantado, la solución de referencia estudiada no se centra en una sola aplicación dentro de la *Difusión Selectiva*, si no que abarca un conjunto de soluciones y aplicaciones.

3.4.2.1 Cartelería Digital (Digital Signage)

Uno de los principales puntos de interés de la *Difusión Selectiva* es la *Cartelería Digital*. A modo de resumen, se presentan los elementos principales de la solución de *Cartelería Digital* ofrecida por la solución de referencia. Ésta está compuesta por varios equipos clave, los cuáles son utilizados también para el resto de aplicaciones de *Difusión Selectiva* dentro de la solución de referencia:



Ilustración 12.-. Elementos principales del servicio de Cartelería Digital de Cisco. Fuente: Cisco 2008-2010.

Se ha de tener en cuenta que es necesario un nuevo *hardware* por cada emplazamiento el cual se ha definido como *DMP*. El *software* del servidor *DMM* es utilizado para proporcionar la planificación y el contenido multimedia los diferentes *DMP*.

Dentro del *DMP* se ha de indicar que no aparece ningún tipo de interactividad asociada al público objetivo y que, por tanto, el tipo de servicio ofrecido es de *Difusión Selectiva Pasiva*. Sin embargo, dentro del *roadmap* (*hoja de ruta*) de la solución de referencia, se ha planificado introducir elementos de adquisición de datos (*sensores*), convirtiendo el servicio en un tipo de *Difusión Selectiva Adaptativa o Activa* (ver apdo. 2.5 Definición de Tipos de Difusión Selectiva).

3.4.2.2 Portal (Web) de Video

Un *Portal Web de Vídeo*⁸ proporciona contenido multimedia en los *Escritorios* de los sistemas de los usuarios suscritos al servicio. La aplicación permite compartir, visualizar e incluso editar videos entre los diferentes usuarios. Los elementos esenciales de la arquitectura para ofrecer un servicio de *Portal Web de Video*, dentro de la arquitectura de referencia, se muestra en la siguiente figura:



Ilustración 13.-. Elementos principales del servicio de Portal de Video de Cisco. Fuente: Cisco 2008-2010.

El servicio de *Portal Web de Video*⁹ propuesto por el fabricante de referencia ofrece la posibilidad de visualizar contenido multimedia a través de una página web. El contenido que se ofrece a cada usuario puede ser selectivo por determinada información configurada en el sistema e incluso puede ser contenido emitido en directo.

Este servicio ofrece al usuario elegir el contenido a visualizar de forma interactiva a partir de listas de contenido multimedia pre-configuradas. En este caso, no es necesario proveer a la solución de *DMPs* que ofrezcan el contenido a un medio de presentación (*pantallas TFT, LCD, monitores, video-wall, proyectores, etc.*). La aplicación presenta el contenido a través de navegadores web y, por tanto, se presenta este contenido mediante el ordenador personal en el escritorio del usuario final suscrito al servicio. Puede establecerse la selección de las *listas de reproducción de contenido* de acuerdo al perfil del usuario que accede al sistema, por lo que se considera una aplicación de *Difusión Selectiva*. El fabricante de referencia utiliza, además,

⁸ En recientes consultas (enero de 2011) a la página de referencia (<http://www.cisco.com>) se ha renombrado la aplicación como *Cisco Show a Share* y aunque, se aproxima a un método de colaboración similar a una red social, se sigue considerando como *Difusión Selectiva* debido a que el acceso se establece según a un criterio: el perfil del usuario registrado.

⁹ Un servicio de *Portal de Video* de acceso público muy popular es el ofrecido por *Google* a través de *youtube*. Dentro del *Portal de Video* de la solución de Referencia se prevén aplicaciones más corporativas difiriendo en algunos aspectos en estos servicios de los de carácter público.

toda la arquitectura utilizada para el resto de aplicaciones (*DME, DMM*) para proveer este servicio a una organización.

3.4.2.3 Sistema de TV Corporativo (*Enterprise TV*)

El último gran servicio de *Difusión Selectiva*, que proporciona la arquitectura y el sistema de referencia, es el de un *Sistema de TV Corporativo* que permite mostrar uno u otros canales dependiendo de ciertos parámetros de configuración.

A continuación (Ilustración 14), se presentan los elementos más importantes para proporcionar el servicio de *Sistema de TV Corporativo*.



Ilustración 14.-. Elementos principales del servicio de Sistema de TV Corporativo de Cisco.
Fuente: Cisco 2008-2010.

Se puede comprobar que los elementos más significativos coinciden con los necesarios para proveer *Cartelería Digital*; sin embargo, el reproductor debe ser diferente, ya que se debe contar con el modelo *4400G* para proveer la funcionalidad de **mando a distancia**. De esta forma se provee interactividad con el elemento de presentación de contenido, pudiendo elegir el usuario el *Canal de TV* (difundido a través de *Multicast*) de la emisión, de modo que se establece una *Arquitectura de Difusión Selectiva Interactiva*, ya que aparece un nuevo medio (mando a distancia) para intercambiar información entre el sistema y el usuario.

Además, es reseñable el hecho de que ofrecer un *Sistema de TV Corporativo* necesita la emisión de varios canales con la característica de 'seleccionables' a través del mando a distancia ya que es una funcionalidad muy demandada. Con el mando a distancia lo que se establece es la elección de los flujos *Multicast* que se están emitiendo en la red a la que el *DMP* está conectado. Otra alternativa es prescindir del mando a distancia emitiendo flujos *Multicast* que son recibidos en el *DMP*; en este último caso, la configuración que se establece en el *DMP* no es variable y se establece un sistema parecido al de *Cartelería Digital*, pero recibiendo flujos *Multicast* con la posibilidad de recibir emisión en directo: el modelo *DMP 4305G* también permite esta funcionalidad.

3.4.3 Arquitectura Funcional de Referencia

A continuación (Ilustración 15), se presenta un resumen de los elementos lógicos principales dentro de la arquitectura ofrecida por Cisco. El escenario inicial se muestra en la Ilustración 15:

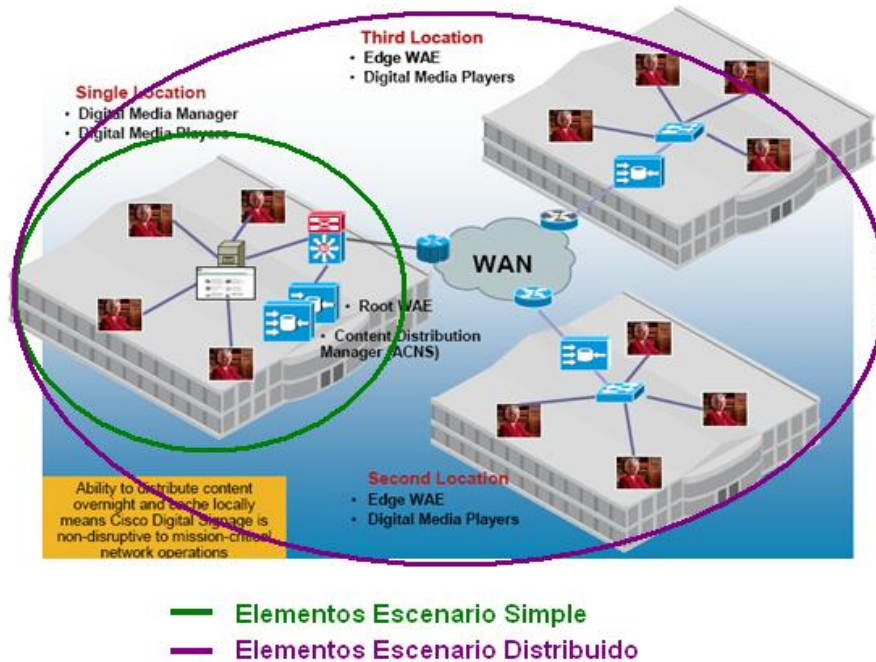


Ilustración 15.-. Arquitectura propuesta por la solución de Cisco (Ejemplo para Cisco Digital Signage). Espacio de Funcionamiento Simple y Espacio de Funcionamiento Distribuido. Fuente: Cisco 2008-2010.

Se puede observar que la arquitectura se presenta con elementos distribuidos, pero con una gestión centralizada. Es decir, supone un contexto *multi-sede* interconectado por una *Red de Área Ampla* (WAN).

Observando la Ilustración 15, se han identificado diferentes equipos y diferentes escenarios no excluyentes. Se ha hecho esta división para intentar resumir la *Arquitectura de Referencia*. Se ha identificado un **Espacio de Funcionamiento Simple**, un **Espacio de Funcionamiento Distribuido** y un **Espacio de Funcionamiento Híbrido** que, aunque el fabricante no lo ha definido así, se espera que resulten apropiados para explicar la arquitectura, para posteriormente poder hacer una comparación con el *Sistema de Difusión Selectiva Diseñado y Desarrollado* (ver Comparativa: Arquitectura Funcional Implementada y Arquitectura Funcional de Referencia).

En la Ilustración 15 se pueden observar algunos elementos importantes para la *Arquitectura Funcional de Referencia*: Root WAE y Edge WAE. Estos dispositivos WAE (*Wide Area Acceleration Engine*), forman parte de la tecnología del servicio WAAS (*Wide Area Application Services*) proporcionado por el fabricante de referencia. Así, la tecnología WAAS de Cisco, permite optimizar el uso del *ancho de banda* de los enlaces WAN de una organización mediante una serie de actuaciones [60]:

- Almacenamiento local de contenido previamente transmitido por los enlaces WAN (caché)
- Eliminación de la redundancia existente y compresión de los datos que se envían por la WAN

- Pre-almacenamiento de contenido en los dispositivos WAEs de las sedes remotas
- Reducción del exceso de tráfico que caracteriza a TCP y optimización del protocolo TCP.

Los WAE, físicamente, pueden ser de dos tipos:

- *Network Module* para un *router*: varían según el modelo (*NME-302*, *NME-502* y *NME-522*) en cuanto a capacidad de almacenamiento y de memoria RAM. Se utilizan sobre todo para los dispositivos con rol *Edge WAE* de las sedes remotas.
- *Appliance*: existen diferentes modelos (*WAE-512*, *WAE-612*, *WAE-674*, *WAE-7341*, y *WAE-7371*). Son auténticos servidores y se suelen instalar en sedes remotas grandes y en la sede central donde se ubica el centro de datos.

Estos dispositivos (WAE) permiten gestionar el *ancho de banda* de forma eficiente y según la complejidad del escenario la configuración [61] será diferente adquiriendo diferentes nombres. Dependiendo del rol que se le asigne se establecerá una configuración u otra.

En la Ilustración 15 también aparece un nuevo elemento que no se ha nombrado hasta ahora: el *ACNS* (*Cisco Application and Content Network System*). Este *software* permite controlar la publicación del contenido multimedia de forma remota [62] desde un mismo punto estableciendo un uso eficiente del *ancho de banda*. La solución *ACNS* facilita la distribución de contenido en redes con limitaciones de *ancho de banda*, debido a que reduce la congestión almacenando y distribuyendo el contenido en el extremo de la red. El *software* *ACNS* combina las tecnologías de distribución del contenido de la red, con la carga previa de contenido; es decir, la solución se ejecuta en combinación con los *Wide Area Application Engines* (WAEs), *Content Distribution Manager* (CDM) y *routers* que soportan encaminamiento de contenido mediante el protocolo *WCCP*. Así, estos componentes de *hardware* y *software* ofrecen:

- *Distribución de contenido en el extremo*: las organizaciones pueden utilizar el dispositivo WAE o el *módulo de red WAE* (*NME-WAE*) con el fin de evitar la congestión de la WAN al almacenar y distribuir el contenido en los equipos que se encuentran en el extremo de la red.
- *Funciones de administración centralizada de contenidos*: un usuario administrador puede utilizar el dispositivo CDM, así como las funciones de administración de la red de forma global, con el fin de administrar y controlar la distribución del contenido multimedia, desde un emplazamiento central.
- *Funciones de enrutamiento de contenido*: las organizaciones pueden utilizar los dispositivos router que soportan encaminamiento de contenido, para el enrutamiento *HTTP* mediante el protocolo *WCCP* (*Web Cache Control Protocol*), con el fin de optimizar la obtención de contenido por parte de los equipos terminales.

Dependiendo del modo de funcionamiento implementado, también aparecen servidores de terceros que proporcionan *Streaming Media* y *Pseudo Streaming* y que se integran dentro de la solución. Estos equipos terceros no se han presentado en la Ilustración 15.

En la Ilustración 16, la cual debe considerarse independientemente del modo de funcionamiento, se muestra un esquema en secuencia de la provisión y configuración del contenido multimedia hasta al alcanzar a los equipos terminales [59]:

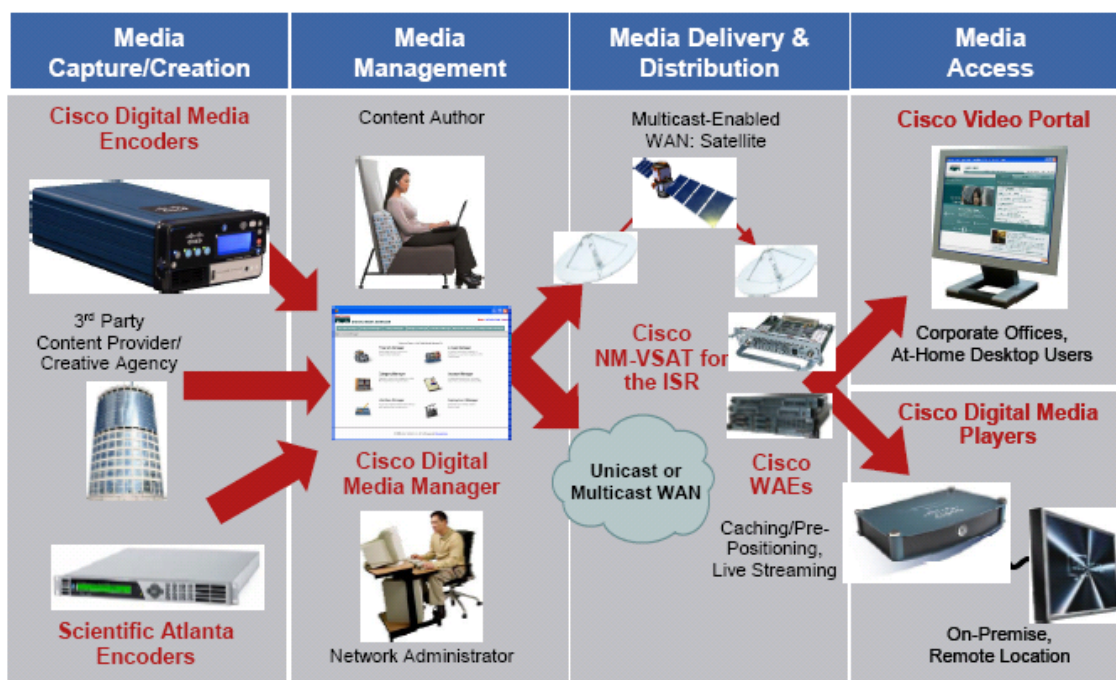


Ilustración 16.-. Elementos que intervienen en la solución distribuida propuesta por Cisco (Cisco Digital Signage Components). Fuente Cisco, 2008-2010.

Si nos centramos en la Ilustración 16, se observa que dentro de la fase de *Administración Multimedia (Media Management)*, habrá dos figuras fundamentales: el *Autor de Contenido (Content Author)* y el *Administrador de Red (Network Administrator)*. El *Autor de Contenido* se encargará de generar y subir al servidor, el material multimedia que se proyectará en las diferentes ubicaciones físicas. Por otra parte, el *Administrador de Red* será el encargado de dimensionar la red y configurar los equipos a través del *DMM*, asignando una distribución *Unicast* o *Multicast*, por ejemplo, o publicando, dentro de la arquitectura, los equipos *WAEs* o los *DMPs*. Se observa también que el fabricante de la arquitectura de referencia presta especial atención a la posibilidad de distribuir el contenido a través de enlaces satélites considerando esta funcionalidad como una característica de valor añadido al sistema.

En los siguientes sub-apartados se realiza la definición de tres tipos de *espacios de funcionamiento* y se mostrará, mediante un ejemplo completo de una de las aplicaciones de *Difusión Selectiva* (el *Portal Web de Vídeo*), cómo intervienen cada uno de los elementos específicos de la *arquitectura general* que se ha presentado en la Ilustración 15. Es importante recalcar de forma inicial, que las aplicaciones que se pueden proporcionar son independientes de si nos encontramos en un *Espacio de Funcionamiento Simple, Distribuido o Híbrido*.

3.4.3.1 Clasificación de los Elementos dentro de la Arquitectura de Referencia

Antes, y para una mejor comprensión de cada uno modos de funcionamiento y ejemplos presentados, se van a definir una serie de conceptos, los cuales nos servirán a la hora de clasificar cada dispositivo que interviene en la arquitectura. Así, clasificamos cada elemento dentro de la *arquitectura de referencia* de acuerdo a su función principal.

Dentro la arquitectura que se ha estudiado se identifican tres conjuntos de elementos o equipos y son utilizados en todas las aplicaciones de *Difusión Selectiva* presentadas. Esta agrupación de los diferentes se establece en:

- **Elementos de Codificación del Contenido Multimedia (Digital Media Encoding).** Estos elementos se configuran para establecerlos como fuente de contenido, principalmente como *Streaming Media*. La función principal de estos equipos es proveer el contenido multimedia, ya sea vídeo bajo demanda (VoD, *Video on Demand*), contenido en directo o emisión de grabaciones.
- **Canal de Administración y Publicación (Channel Administration and Publishing).** Con estos equipos se permite publicar y administrar el contenido en los distintos emplazamientos. Con el acceso a la administración de estos dispositivos se configura el sistema para proporcionar la distribución del contenido a los equipos finales que presentan los archivos multimedia. Dentro de este conjunto de equipos se encuentra el servidor que ejecuta la aplicación *DMM*, el servidor que ejecuta el *software ACNS*, así como el módulo de ampliación para proveer los servicios de *Portal Web de Vídeo*, el cual puede estar ejecutando también en un servidor independiente.
- **Elementos de Publicación de Contenido.** Son los elementos que permiten mostrar el contenido multimedia a la audiencia final. Serán los equipos informáticos para el caso de *Portal Web de Vídeo* o las pantallas de presentación (TFTs, LCDs, monitores, proyectores, video-wall, etc.) en el caso de que se utilicen *DMPs* en *Cartelería Digital* o *Canal de TV Corporativo*.

Los *DMPs* pueden quedar dentro de la definición del conjunto de *Canal de Administración y Publicación* o dentro de los *Elementos de Publicación* dependiendo del contexto (*Espacio de Funcionamiento*) y la aplicación; sin embargo, parece que es más acertado clasificarlos como *Elementos de Publicación de Contenido* ya que estos serán lo que proporcionen de forma adecuada el contenido a un sistema de presentación independientemente de cómo haya sido codificado.

3.4.3.2 Espacio de Funcionamiento Simple

El *Espacio de Funcionamiento Simple* es un esquema donde se encuentran, dentro de la misma red, todos los equipos necesarios para proveer la distribución del contenido multimedia. Nos encontramos dentro de un contexto donde los equipos se encuentran en comunicación local (*LAN*) y en la que no intervienen comunicaciones a través de una *WAN* (siendo este último, el aspecto decisivo que clasifica el *espacio de funcionamiento*). Es decir, aunque puede estar en subredes locales (*LANs*) diferentes separadas por *routers*, no se aplican habilidades para implementar *arquitecturas* que intenten realizar un uso eficiente del *ancho de banda* entre los equipos, debido a que no existe una *WAN*. Es por ello que dentro del contexto del proyecto se supone que no hay necesidad apremiante de optimizar el *ancho de banda* disponible.

A continuación, a través un ejemplo, se muestra el espacio de funcionamiento simple:

Ejemplo Espacio de Funcionamiento Simple: Portal de Vídeo en Difusión Selectiva

El esquema general de este *espacio de funcionamiento* para la aplicación de *Portal de Vídeo* se muestra en el siguiente esquema:

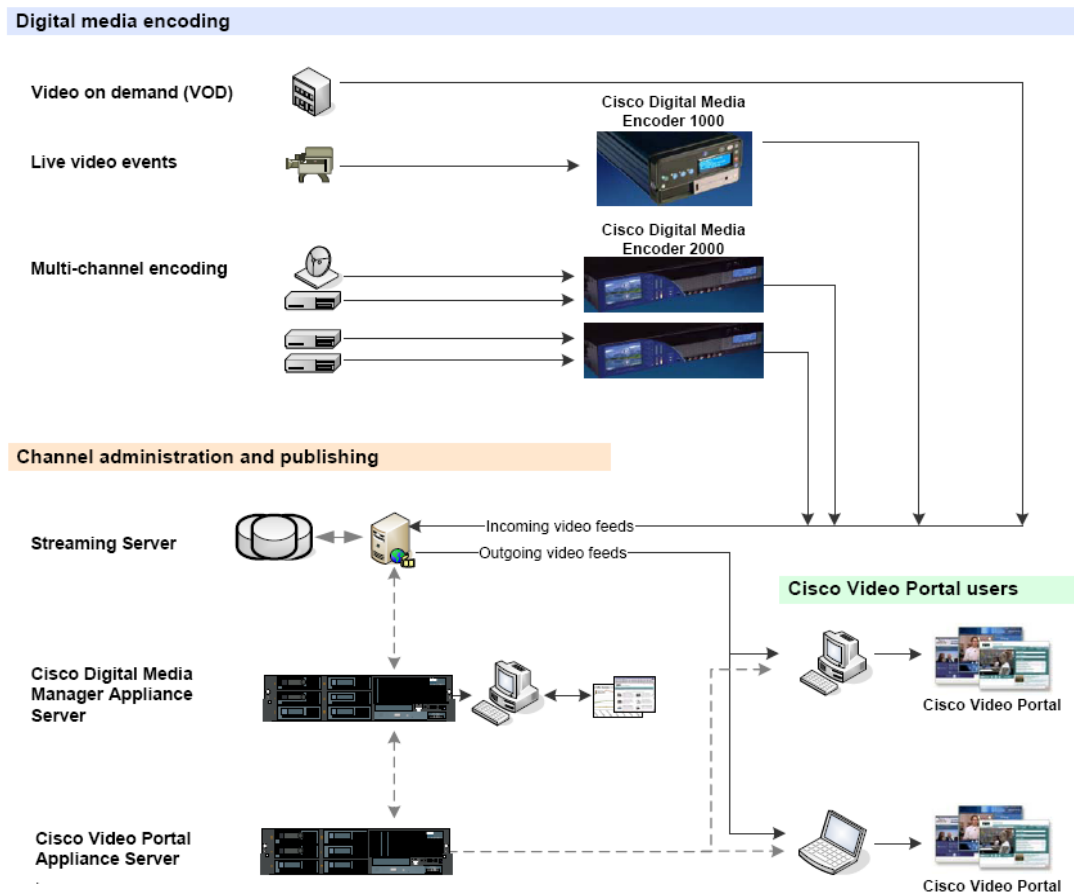


Ilustración 17.-. Portal de Video de Cisco en configuración de Espacio de Funcionamiento Simple en la solución de referencia. Fuente: Cisco 2008-2010.

En la ilustración se puede observar cada uno de los conjuntos de elementos separados en función del cometido de cada uno de los equipos. Esta clasificación se ha definido en el apartado 3.4.3.1: *Codificación del Contenido Multimedia, Canal de Administración y Publicación y Elementos de Publicación (Usuarios del Portal Web de Video)*.

Seguidamente, se hace una descripción de los elementos significativos involucrados en la arquitectura del ejemplo. Dentro del conjunto de equipos que pertenecen a la **Codificación del Contenido Multimedia** tenemos:

- **Video on Demand (VoD):** Son archivos de vídeo, películas, clips... que se tienen almacenados en un *servidor*. Los vídeos son publicado directamente en el servidor *Streaming*.
- **Live Video Events:** Codificación de transmisiones en directo. Estos equipos adquieren señales de vídeo compuesto. El contenido codificado se envía mediante un *DME* al servidor *Streaming* y se publica desde el mismo al resto de elementos.
- **Multi-Channel Encoding:** A través de varios equipos se pueden codificar varios canales para su distribución de forma simultánea. Normalmente este tipo de codificadores tienen dos entradas por dispositivos, ya sean TV por cable o satélite, DVD, VCR,... Estos equipos, además, tienen entradas digitales y analógicas con diferentes modo de conexión: conectores *SDI*, *S-Video* y compuesto por entradas de audio balanceadas y no balanceadas. Una diferencia con los equipos *Live Video Events* es que permiten varias fuentes de entrada, las cuales pueden ser encapsuladas a través de una misma interfaz de red y siguiendo determinados estándares (*multiplexación de video*).

Aparecen también, en el esquema del ejemplo (*Portal de Vídeo*) de *Espacio de Funcionamiento Simple* (Ilustración 17), equipos pertenecientes al **Canal de Administración y Publicación** de contenido. La relación de estos equipos es la siguiente:

- **Servidores de Streaming y Almacenamiento:** Estos servidores reciben datos de vídeo y los vuelven a publicar en los equipos pertenecientes al conjunto de *Elementos de Publicación* a través de técnicas de *Streaming*. Ejemplos de este tipo de servidores son: *Microsoft Windows Media Services* [49], servidor *DSS (Darwing Streaming Server)* [63] o *Wowza Media Server* [64] entre otros.
- **Servidor de la Aplicación DMM (Digital Media Manager):** Esta aplicación se instala directamente en un servidor.
- **Servidor de la Aplicación de Portal de Vídeo:** Es una aplicación que se instala en un servidor independiente. La aplicación de *Portal de Vídeo* permite establecer canales para *vídeo web* y presentar el contenido de forma diferenciada en función de la audiencia. La configuración de este servidor se realiza en el *DMM*, aunque físicamente se plantea como equipo *hardware* independiente, de modo que existe una sincronización entre el *DMM* y el *Portal de Vídeo*.

Mediante la configuración de la aplicación *DMM* permite administrar las *listas de reproducción*, la *presentación* y la *publicación* de nuevo contenido multimedia. La configuración de la publicación de contenido, en todo el sistema, se realiza de forma centralizada a través del *DMM*. Para realizar la configuración, tan sólo es necesario un navegador web genérico que permite la configuración a través de una interfaz web. Por esta razón, se puede realizar la configuración del sistema desde cualquier equipo que disponga de conectividad hacia el servidor *DMM*, aunque no se tenga acceso al resto de equipos desde el equipo que se realiza la configuración. Esta característica permite proveer cierta seguridad adicional al sistema. El *DMM* es un elemento clave dentro de la arquitectura ya que provee una administración centralizada de la mayoría de los equipos.

En el servidor *DMM* se publican, mediante configuración, el servidor *Streaming* disponible y también la interacción con el servidor de la aplicación de *Portal de Vídeo*.

En el servidor *Streaming* se publican los elementos de *Codificación de Contenido Multimedia*. Así, si en la configuración del *DMM* se tiene almacenada, entre otras, una lista de vídeos disponibles del servidor *vídeos bajo demanda (VoD)*, éste se debe registrar en el servidor *Streaming*. De este modo el servidor *Streaming* los ofrecerán directamente al equipo terminal mediante el protocolo establecido según se vayan demandando al seleccionar su reproducción [58][59] [65]. De igual forma que el servidor de *vídeos bajo demanda (VoD)*, tanto los *codificadores de vídeo en directo (Live Video Events)*, como los *codificadores multicanal*, se deben registrar mediante configuración en el servidor *Streaming* y, así, es éste último el que proporciona directamente el contenido a los equipos terminales.

Dentro del *Portal de Vídeo*, la lista de contenido multimedia disponible a cada usuario se puede restringir por perfil, estableciendo de este modo *Difusión Selectiva*. El contenido se muestra directamente en los navegadores web de los equipos clientes a través de una aplicación web que reside en servidor del *Portal de Vídeo*.

3.4.3.3 *Espacio de Funcionamiento Distribuido*

La característica diferenciadora de un *Espacio de Funcionamiento Distribuido*, tal como se especifica en este proyecto, consiste en la existencia de una *WAN* que comunica los *equipos terminales (DMPs)* con el *DMM*. Esta *WAN*, a su vez, tiene un *ancho de banda* limitado y uno de los objetivos iniciales de diseño es preservar este recurso. Por ello, se considera que haya

varios modos de funcionamiento en la *Arquitectura de Referencia*, precisamente porque aparecen nuevos elementos en la arquitectura que permiten utilizar de forma eficiente el *ancho de banda* entre las hipotéticas sedes que están distribuidas geográficamente.

Con el objetivo de realizar un uso eficiente del *ancho de banda* de la WAN, en la *solución de referencia*, se introducen [59] nuevos elementos *hardware*. Éstos se denominan *Root WAE* (*Wide Area Application Engine*) y *Edge WAE* y permiten usar el *ancho de banda* de forma eficiente. También aparece un sistema *software* de gestión de las nuevas capacidades, *Cisco Application and Content Network System (ACNS)*, en el cual se registran los elementos WAEs y permite una configuración apropiada de las sedes remotas, estableciendo también el requisito de escalabilidad (ver Ilustración 15). Por tanto, la escalabilidad en grandes despliegues del DMS, se obtiene añadiendo en el sistema el *software Cisco Application and Content Network System (ACNS)*, complementado por las capacidades de los equipos WAE.

Ejemplo Espacio de Funcionamiento Distribuido: Portal de Vídeo en Difusión Selectiva

La arquitectura del sistema en un *Espacio de Funcionamiento Distribuido* se hace más compleja al aparecer nuevos dispositivos que la forman. Se ha de indicar que además de las técnicas que se proponen, se puede hacer uso de otras complementarias, como son la compresión de datos sobre la WAN a través de *Cisco Wide Area Application Services (WAAS)*[66]. Sin embargo, dentro de la solución no se propone un mecanismo concreto por lo que no se ha incluido en el ejemplo.

A continuación, se muestra el esquema de la *arquitectura de referencia* para el ejemplo de la aplicación *Portal de Vídeo*; en el cual, también se separan gráficamente, en la ilustración, cada conjunto de equipos cuyas tareas son diferentes: *Codificación del Contenido Multimedia*, *Canal de Administración y Publicación (con ACNS y múltiples dispositivos WAEs)* [61] y *Elementos de Publicación (Usuarios del Portal de Video)*

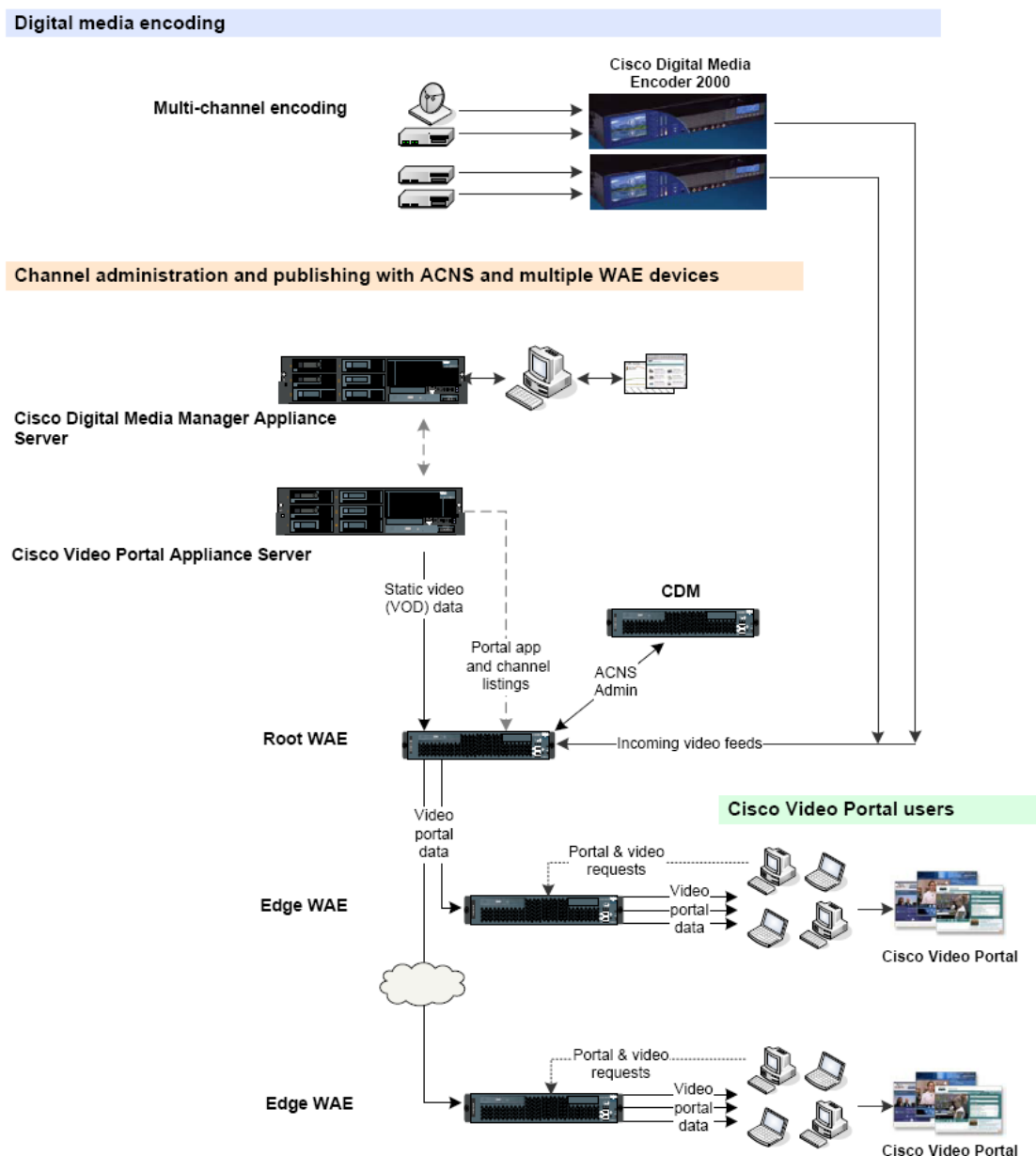


Ilustración 18.-. Portal de Video de Cisco en configuración de Espacio de Funcionamiento Distribuido en la solución de referencia. Fuente: Cisco 2008-2010.

Con respecto al *Espacio de Funcionamiento Simple*, en el *Espacio de Funcionamiento Distribuido*, se han identificado nuevos elementos de referencia adicionales dentro del *Canal de Administración y Publicación*:

- **Content Distribution Manager (CDM):** Es el servidor que tiene instalada la herramienta ACNS. Con esta herramienta *software* se configura la lógica de cómo se va distribuir el contenido multimedia.
- **Root WAE:** Es un equipamiento hardware que suele ir instalado como una tarjeta adicional en los routers. Funcionalmente, recibe datos de vídeo de otros servidores y codificadores y los distribuye a los diferentes *Edge WAE* que sirven los datos multimedia a los usuarios directamente a través de los *Elementos de Publicación* (en el ejemplo los usuarios de *Portal Web de Video*).

- **Edge WAE:** Funcionalmente reciben los vídeos desde el *Root WAE* y los vuelve a transmitir a los *Elementos de Publicación* que están configurados en el *Portal Web de Vídeo*. Estos equipos, los *Edge WAE*, funcionan de forma similar a un *Proxy* dentro de la arquitectura.

En el *CDM* se publican los distintos *dispositivos WAE* disponibles en la arquitectura y, a su vez, éste establece comunicación con el *root WAE*, el cual es el dispositivo que realmente interactúa con el resto de equipos *WAEs* (*Edge WAEs*). Como se puede observar en la Ilustración 18, las peticiones de los equipos terminales se realizan a través de los distintos *Edge WAE*.

El *Edge WAE* funcionará como un *proxy* dentro de cada sede. Para el correcto funcionamiento del mecanismo, no es necesario configurar los *Edge WAE* como *proxy* en los diferentes *DMPs*; sino que en el elemento pasarela de la *WAN* de cada sede, se debe configurar el protocolo *Web Cache Control Protocol* o *Web Cache Communication Protocol* (*WCCP*). El *WCCP* reencamina las peticiones (en realidad el tráfico *IP* u otros protocolos) a los *Edge WAE*. De esta forma aparece un *proxy transparente*, a modo de portal cautivo, que permite mejorar la eficiencia del *ancho de banda*.

Para poder incluir *vídeo bajo demanda* (*VoD*) se ha de incluir un servidor de *Streaming* intermedio. A continuación se muestra un esquema de la variante del *Espacio de Funcionamiento Distribuido* en el que se incluye también la posibilidad de incluir *VoD*, del mismo modo que se ha descrito en el apartado 3.4.3.2 (*Espacio de Funcionamiento Simple*).

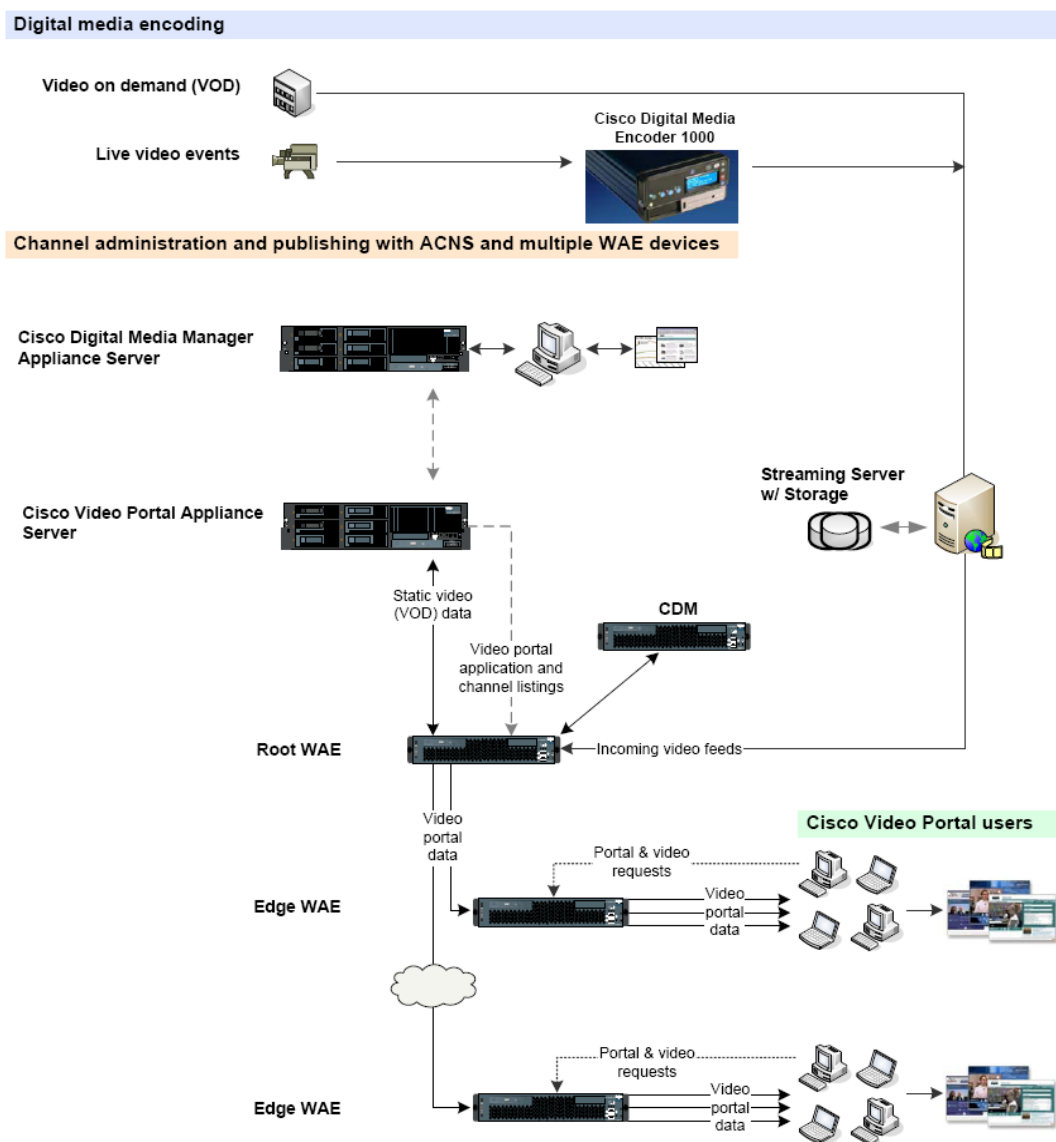


Ilustración 19.- Portal de Video de Cisco en configuración de Espacio de Funcionamiento Distribuido en la solución de referencia con capacidad para proporcionar VoD. Fuente: Cisco 2008-2010.

El *Servidor de Streaming y Almacenamiento* se debe introducir entre el *Root WAE* y los elementos que forman la *Codificación de Contenido Multimedia*. El *Servidor de Streaming y Almacenamiento* es necesario para proporcionar al sistema la posibilidad de ofrecer *video bajo demanda (VoD)* y no sólo flujos de codificación multicanal como se representaba la Ilustración 18. Por tanto, como se representa en la Ilustración 19, el *Servidor de Streaming con Almacenamiento (Streaming Server w/ Storage)* es utilizado como pasarela ante el *Root WAE*.

Se ha de considerar que esta configuración funcional no es la única alternativa para proporcionar *Video bajo Demanda (VoD)* y se permite otras configuraciones. Por ejemplo, otra alternativa, es introducir el *Servidor de Streaming y Almacenamiento* como el elemento que proporciona el *video bajo demanda* directamente al *Root WAE*, pero al mismo nivel que el resto de dispositivos clasificados como de *Codificación de Contenido Multimedia (Codificador Multicanal y los Eventos en Directo)*. Un ejemplo de esta configuración es el *Espacio de Funcionamiento Híbrido* que se muestra a continuación.

3.4.3.4 *Espacio de Funcionamiento Híbrido*

Se define en el contexto del proyecto fin de carrera el *Espacio de Funcionamiento Híbrido*, como la configuración que mantiene la distribución de equipos inicial del *Espacio de Funcionamiento Distribuido*, pero manteniendo el contexto de operación del *Espacio de Funcionamiento Simple* en el que los equipos remotos hacen las peticiones al mismo equipo; en este caso al *Standalone WAE* que tendrá el rol de equipo *proxy* de contenido.

Es decir, ya sea en la configuración del servicio *Portal de Vídeo*, *Cartelería Digital* o *Sistema de TV Corporativo*, las peticiones irán dirigidas al equipo *Standalone WAE*; por lo que aunque hubiera una WAN intermedia en el camino hasta el *Standalone WAE* no se estarían realizando operaciones para utilizar de forma eficiente el *ancho de banda* de esta WAN si el *DMP* o el equipo terminal que realiza la petición está en una sede remota.

En este modo de funcionamiento, el *software ACNS* y, por tanto, el servidor que ejecuta esta aplicación, no es necesario debido a que no se controlan diferentes elementos *WAE* distribuidos. No obstante, este tipo de *espacio de funcionamiento* es una elección muy conveniente en una primera implementación de la arquitectura. Así, añadir sedes remotas separadas mediante una WAN a un *Espacio de Funcionamiento Híbrido*, se hace mucho más sencillo debido, precisamente, a que tan sólo hay que integrar el *ACNS* y los nuevos *edge WAEs* (pasando el *Standalone WAE* a tener el rol de *Root WAE*). Esto se traduce en ahorro de costes al modificar el sistema lo menos posible. Indicar que, el *Standalone WAE*, permitiría realizar la distribución a través de *Unicast* o *Multicast* y, en el caso de propagar el contenido de forma *Multicast*, puede proporcionar eficiencia en determinados servicios como es el caso de un *Sistema de TV Corporativa*.

Ejemplo de Espacio de Funcionamiento Híbrido: Portal de Vídeo en Difusión Selectiva

La arquitectura que se ha definido en el proyecto fin de carrera para el fabricante de referencia como *Espacio de Funcionamiento Híbrido*, se muestra en la siguiente ilustración; donde la clasificación de los elementos se sigue estableciendo en tres grupos: *Codificación del Contenido Multimedia*, *Canal de Administración y Publicación* y *Elementos de Publicación (Usuarios del Portal de Vídeo)*.

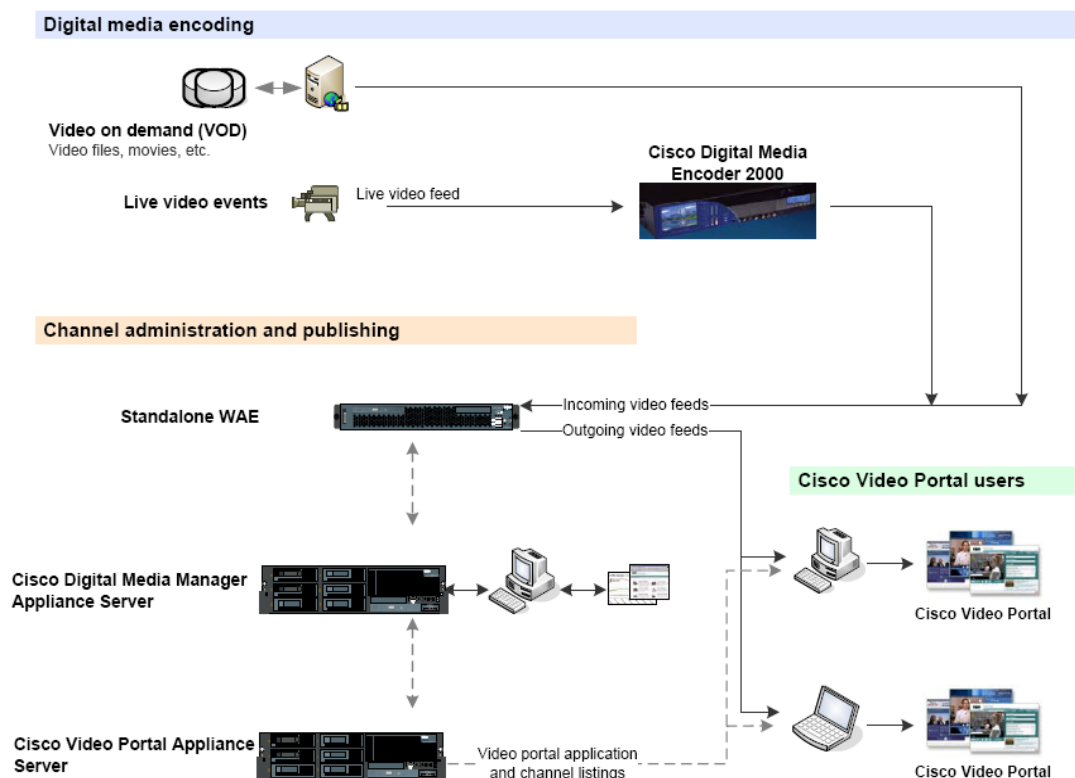


Ilustración 20.-. Portal de Video de Cisco en configuración de Espacio Híbrido en la solución de referencia.
Fuente: Cisco 2008-2010

La arquitectura tiene elementos comunes a la del *Espacio de Funcionamiento Simple* y a la del *Espacio de Funcionamiento Distribuido*.

Básicamente el esquema y modo de funcionamiento es similar al del *Espacio de Funcionamiento Simple*; no obstante, aparecen nuevos roles característicos del *Espacio de Funcionamiento Distribuido* que se ha definido. Un nuevo elemento se define en el *Canal de Administración y Publicación*, el dispositivo es el siguiente:

- **Standalone WAE:** Es el mismo *hardware* que puede ser utilizado como *root WAE*. Recibe los vídeos desde los dispositivos que forman parte de la *Codificación del Contenido Multimedia*, tanto para contenido en directo como para codificación multicanal de contenido y vídeo bajo demanda (VoD), y vuelve a publicar los datos a los equipos que forman parte de los *Elementos de Publicación*, en el ejemplo, los *usuarios del Portal (Web) de Video*. Sería similar, por posicionamiento dentro de la arquitectura, al *Root WAE*. Sin embargo, realiza también las funciones del *Edge WAE* al proporcionar, este dispositivo, directamente el contenido a los *Elementos de Publicación*.

La primera característica que se aprecia es que se puede formar con el mismo número de dispositivos que en el *Espacio de Funcionamiento Simple* y que no aporta ventajas explícitas en principio con respecto a este modo de funcionamiento. La diferencia más significativa es que el *Standalone WAE* realiza las funciones del *servidor de Streaming* del *Espacio de Funcionamiento Simple*, en el sentido de que proporciona los datos directamente a los *Elementos de Publicación*. Dentro de la arquitectura de *Espacio de Funcionamiento Distribuido* esta función la realizan los diferentes *Edge WAE*. Se ha de apreciar, que también aparece en *Servidor de Streaming* en el *Espacio de Funcionamiento Híbrido*, pero es para proporcionar vídeo bajo

demanda (VoD) al *Standalone WAE*, característica ésta, común a la arquitectura de *Espacio de Funcionamiento Distribuido*.

Por otra parte, no existe *CDM* (*Content Distribution Manager*) para configurar la distribución de contenido debido a que todas las peticiones se canalizan a través del *Standalone WAE* y el software *ACNS*, como se ha comentado, sirve para configurar los elementos distribuidos del sistema (los *Edge WAEs*) y mejorar el uso de la WAN.

3.5 Arquitectura de Difusión Selectiva de Referencia Completa

En éste apartado vamos a proporcionar una visión de la arquitectura de referencia completa del sistema de *Difusión Selectiva*, usando un esquema de *Espacio de Funcionamiento Distribuido*, pero mostrando el resto de las aplicaciones y no sólo la que se ha mostrado de ejemplo (*Portal de Vídeo para Difusión Selectiva*) en los apartados previos.

Dentro del esquema del sistema aparecen los tres servicios o aplicaciones consideradas: *Cartelería Digital*, *Sistema de TV Corporativo* y *Portal de Vídeo*, bajo la misma *arquitectura funcional*.

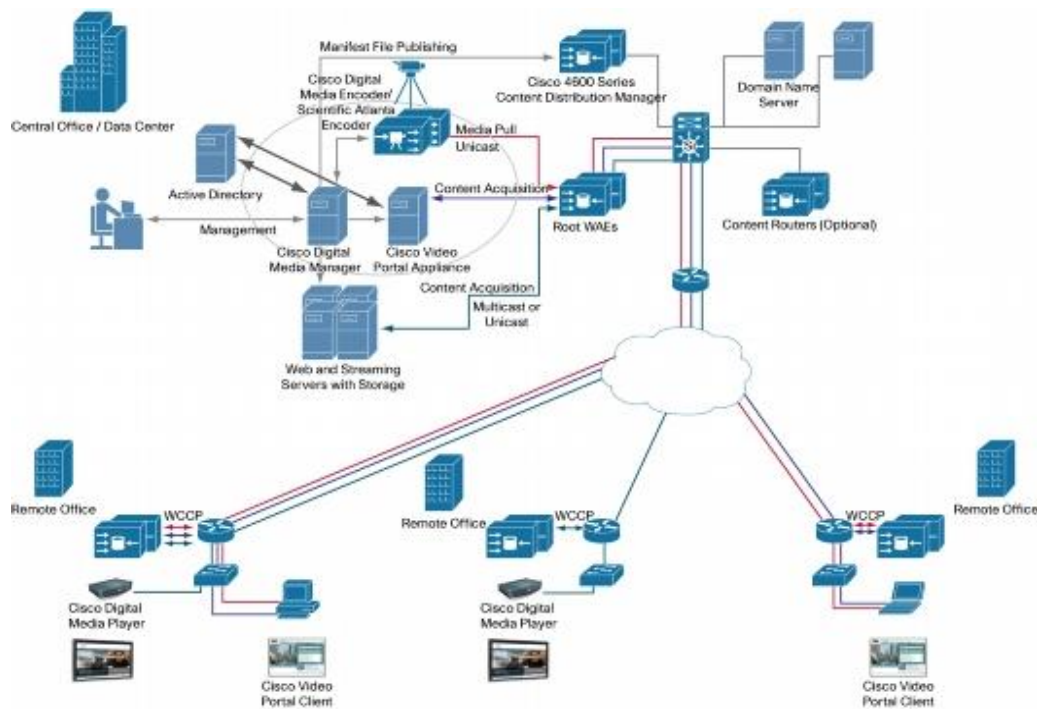


Ilustración 21.-. Resumen de arquitectura de Difusión Selectiva de fabricante referencia. Fuente: Cisco 2008-2010.

Algunos dispositivos que aparecen en la Ilustración 21 son elementos que proporcionan otras funciones definidas en la arquitectura, estos dispositivos son el *Directorio Activo* (*Active Directory*) y los servidores de dominio (*DNS, Domain Name Server*).

El *Directorio Activo* (*Active Directory*) aparece en el sistema debido a que la *arquitectura de referencia* permite, en la configuración del *DMM*, establecer usuarios con acceso restringido a funcionalidades, o niveles, y estos usuarios pueden ser validados, por ejemplo, mediante el *Directorio Activo* de forma centralizada; no obstante, este elemento es típico también, como es evidente, en otras aplicaciones diferentes de la *Difusión Selectiva* (ver Ilustración 21).

Se observa en la Ilustración 21 la presencia de la utilización del *Web Cache Control Protocol* o *Web Cache Communication Protocol (WCCP)*. El WCCP es un protocolo definido por Cisco que se emplea para definir la manera de redirigir el tráfico IP desde un *router* a un *proxy-caché*.

En la arquitectura del sistema las funciones de *proxy-caché* corresponden a los *Edge WAE* definidos en la arquitectura de referencia a lo largo de los ejemplos (ver 3.4.3 Arquitectura Funcional de Referencia). Por tanto, en la *arquitectura de referencia* la configuración de los *Elementos de Publicación (Clientes del Portal de Video y DMPs)* establece que las peticiones vayan dirigidas o bien al *Root WAE* o a los *Servidores de Contenido*. Sin embargo, el flujo de determinados protocolos (*HTTP*), es capturado por el router con el protocolo WCCP, a modo de portal cautivo, lo que produce que la petición se realice en realidad sobre el *Edge WAE* de la sede donde se encuentra el *Elemento de Publicación*.

A continuación (Ilustración 22), se muestra un ejemplo del flujo de los protocolos a través de la WAN en la *arquitectura funcional de referencia*:

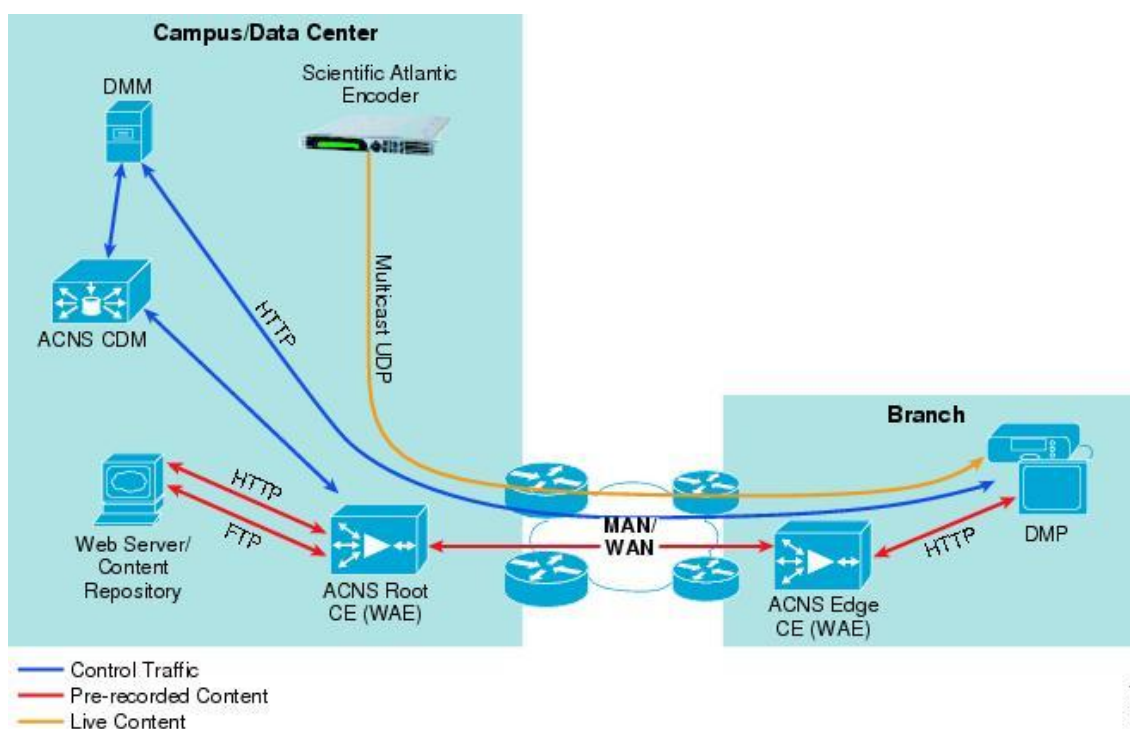


Ilustración 22.- Ejemplo de uso de Protocolos de comunicación entre los diferentes elementos de la arquitectura en un Espacio de Funcionamiento Distribuido. Fuente: Cisco 2010

Como se observa en la Ilustración 22 el tráfico *Multicast UDP (Live Content, Contenido en Directo)* se encamina a través de la WAN, desde la sede principal a la sede remota, por lo que los *routers* deben soportar, además de establecer una configuración especial, el encaminamiento del tráfico *Multicast*.

El protocolo WCCP permite capturar las peticiones del contenido multimedia (*HTTP*) de los *equipos terminales* al repositorio de contenido y redirigirlas al *Edge WAE*. El *Edge WAE* suministra el contenido multimedia directamente a los *equipos terminales*, debido a que tiene almacenado el contenido de forma local. Anteriormente, el *Root WAE* ha almacenado sobre el *Edge WAE* una copia del contenido según la configuración del ACNS, por lo que, independientemente de número de *DMPs* o usuarios de la *aplicación Portal de Video*, se realizará un solo flujo para suministrar el contenido multimedia a través de la WAN, comportándose el *Edge WAE* como un elemento *proxy-caché* [66].

Se observa en la Ilustración 22, también, que el *DMM* establece flujos de control mediante el protocolo *HTTP* a los *equipos terminales (DMPs)* y al *Root WAE*. Estos flujos representan el intercambio de la configuración entre los equipos.

3.6 Resumen

En este capítulo se han presentado las características más importantes de los sistemas de *Difusión Selectiva* estudiados. Se han encontrado soluciones de muy diversas características y que implementan *arquitecturas funcionales* muy distintas. Con todas las soluciones estudiadas se han definido las características más importantes de las arquitecturas: *desarrolladas para propósito específico, propósito general, cliente-servidor, distribuidas, basadas en software de aplicación, basadas en tecnologías web, diseños de equipos de hardware genérico, hardware específico o hardware integrado*.

Con las características identificadas se ha propuesto una evolución de las arquitecturas encontradas que permite clasificar la madurez de una solución concreta. Se puede concluir, aunque no se ha tratado dentro del proyecto proponer la definición del *hardware* de los equipos, que para el desarrollo de una solución inicial la mejor opción quedaría en la elección de *hardware integrado*, al menos para el diseño de los *equipos terminales* ya que obtienen parte de las ventajas de ambos sistemas: soportan los sistemas operativos de uso común, tienen un consumo reducido, mantienen las facilidades en el desarrollo de aplicaciones, permite una integración con diferentes sensores razonable y tienen un coste menor que el *hardware* al que se ha denominado *hardware genérico*.

Se ha seleccionado la solución desarrollada por *Cisco*, de entre todas las soluciones encontradas, por ser una solución evolucionada y con unas características que nos permite considerarla como una solución de referencia. La solución propuesta por *Cisco*, se denomina *Digital Media System* o *Digital Media Suite (DMS)* y es una solución que trata el problema de la *Difusión Selectiva* como un problema general, en lugar de como un problema concreto enfocado a la *Cartelería Digital*.

La solución *DMS* permite varios modos de funcionamiento, cuya característica que hemos marcado como diferenciadora es la presencia o no de una *WAN* entre el equipo principal y los equipos terminales (*DMP, Digital Media Player*). Estos modos de funcionamiento los hemos denominado *Espacio de Funcionamiento Simple, Espacio de Funcionamiento Distribuido y Espacio de Funcionamiento Híbrido* y según cuál estudiemos estarán presentes unos equipos u otros en su *arquitectura funcional*.

Hemos identificado en la solución de referencia una serie de características:

- Es una solución de propósito general
- Soporta difusión de contenido en directo, tanto *Unicast*, como *Multicast*
- Provee acceso a archivos de audio y video bajo demanda
- Reduce el impacto del *ancho de banda* de video en la red mediante estrategias y protocolos que permiten utilizar el ancho de banda de forma eficiente
- Permite separar y priorizar el tráfico en la red *WAN* a través de otras tecnologías y desarrollos del propio fabricante de referencia

La visión general de las arquitecturas y soluciones comerciales, así como el estudio en profundidad de la solución de referencia, ha sido muy útil a la hora de tomar decisiones y abordar el diseño de un *Marco de Aplicaciones para Difusión Selectiva*, concluyendo que hay una relación directa con la *arquitectura funcional* esperada.

PARTE III. Diseño y Desarrollo del Marco

Capítulo 4 Resumen del Sistema de Difusión Selectiva Implementado y Arquitectura Funcional Definida

En este capítulo se presenta la *arquitectura funcional definida e implementada* en este proyecto fin de carrera. No referimos como *arquitectura funcional* al funcionamiento e interacción de los distintos equipos que intervienen en el sistema. Por otra parte, el *Marco* hace referencia a los elementos desarrollados (*scripts*, aplicaciones desarrolladas, instalación de aplicaciones necesarias y configuraciones) y el método de interacción y comunicación entre estos elementos, para llevar a cabo el funcionamiento de cada equipo dentro de la *arquitectura funcional*. Por tanto la *arquitectura funcional* es una parte fundamental del *Marco para las Aplicaciones de Difusión Selectiva* implementado ya que, según se diseñe, establecerá la lógica de las aplicaciones y configuraciones que deberán realizarse en el diseño y desarrollo del *Marco*.

En este capítulo vamos a cubrir los siguientes aspectos en relación a la *arquitectura funcional implementada*:

- Enumerar los requisitos que se han considerado necesarios en la *arquitectura funcional*.
- Definir la terminología y conceptos, que se adapten explícitamente a los elementos que forman la *arquitectura funcional* que se ha definido. Es importante comprender cada una de los términos aquí planteados, ya que se utilizarán en el resto de capítulos.
- *Espacios de Trabajo* definidos en la *arquitectura funcional implementada* y diferencias con los *Espacios de Funcionamiento* definidos en la *arquitectura funcional de referencia*.
- Explicar, *grosso modo*, el funcionamiento completo del sistema
- Definir el concepto de *arquitectura funcional Auto-Proxy* dentro del *Marco* desarrollado. El desarrollo de la *arquitectura funcional Auto-Proxy*, será el mecanismo utilizado para salvaguardar el *ancho de banda* WAN.
- Mostrar otras alternativas de desarrollo de arquitecturas funcionales tipo *proxy*.
- Exponer las consideraciones de la *arquitectura funcional definida* que están íntimamente relacionada con el desarrollo del *Marco*: necesidad de la sincronización de hora entre los equipos y aspectos relacionados con la seguridad.

4.1 Requisitos Funcionales de la Arquitectura

Los requisitos básicos, que deberá cumplir la *arquitectura funcional* para posteriormente desarrollar el *Marco*, son los siguientes:

1. *Arquitectura Cliente-Servidor*: el factor más importante, a la hora de desarrollar el *Marco* con una *arquitectura funcional Cliente-Servidor*, es la necesidad de tener una gestión centralizada de todas las programaciones de todos los medios de presentación disponibles en el sistema de difusión selectiva. Se ha podido comprobar, a partir de varios ejemplos de las soluciones consultadas [67] [68] (Tabla 4, Capítulo 3), que es la solución más aceptada, incluso para las soluciones que utilizan mecanismo de distribución de contenido descentralizado [69]. En una arquitectura *Cliente-Servidor*,

- desde un servidor principal se podrá aplicar la configuración de la programación del contenido multimedia a todos los equipos.
2. *Capacidad de Trabajar con Alta Dispersión Geográfica:* la difusión de contenido en función del emplazamiento físico es uno de los discriminadores más utilizados en la *Difusión Selectiva*, por lo que es necesario disponer, al menos, de algún parámetro geográfico que diferencie el tipo de contenido multimedia a difundir. Esta dispersión, normalmente incluye una WAN que conecta los diferentes equipos.
 3. *Necesidad de Preservar el Ancho de Banda:* debido a la dispersión geográfica de los *equipos terminales* y, por tanto, de la posible presencia de una WAN, se ha de hacer un uso eficiente del *ancho de banda* utilizado para la difusión del contenido multimedia. Esto es debido a que el *ancho de banda* suele ser un recurso caro y, además, suele ser un factor crítico para otras aplicaciones de las organizaciones.
 4. *Establecer Independencia de la Conectividad de Red entre los Diferentes Elementos que Intervienen en el Marco.* El tipo de conexión no debe ser un impedimento a la hora de implantar el sistema: ya sea ésta xDSL, RDSI, 3G, VPN,... el sistema no se debe ver afectado por el tipo de conexión en una situación estable de funcionamiento.

4.2 Conceptos de la Arquitectura Funcional Definida e Implementada

4.2.1 Terminología Específica en la Arquitectura Diseñada

Antes de seguir con la definición de la *arquitectura*, necesaria para diseñar y desarrollar el *Marco para Aplicaciones de Difusión Selectiva*, se van a definir una serie de términos concretos, que se han introducido dentro del presente proyecto, para referirnos a los diferentes entes de la *arquitectura funcional definida*.

A continuación se define la terminología específica, la cual será utilizada para referirnos a los diferentes elementos en el presente documento:

Ubicación.-. Será una configuración específica y, a dicha configuración, se le dará un nombre concreto. Los equipos subscritos a una determinada *Ubicación* emitirán el contenido multimedia programado para un *Escenario* concreto. La *Ubicación* irá asociada normalmente a un lugar físico; por ejemplo, si estuviéramos implantado el sistema en la *Universidad Carlos III de Madrid*, una *Ubicación* podría denominarse '*Biblioteca*' y otra '*Cafetería*'; indicando, con el propio nombre, que estas configuraciones se emitirán en dichos emplazamientos o ubicaciones físicas. El detalle de la configuración de las diferentes *ubicaciones* se puede consultar en el Anexo C. Manual del Usuario Administrador (UBICACIONES >> Seleccionar/Crear Ubicación).

Escenario.-. Será una de las infinitas posibles configuraciones espaciales de una *Ubicación*. El *escenario* no se refiere a las *listas programadas*, el cual corresponde al plano temporal, sino que se refiere a la definición de *zonas* y fondo de pantalla que complementarán la forma de presentar el contenido multimedia. Así, un *Escenario* estará dividido en varias *zonas* donde se podrán emitir diferente contenido multimedia a modo de *listas de reproducción*. Como decisión de ingeniería se ha pensado que por cada *Ubicación* se tengan hasta cuatro posibles *escenarios* pre-configurados; el detalle de la configuración de los *escenarios* se puede consultar en el Anexo C. Manual del Usuario Administrador (UBICACIONES >> Configurar Ubicación >> Zonas a Esc).

Zonas.-. serán secciones rectangulares que se podrán posicionar sobre la pantalla de presentación de contenido. Cada *Zona* de cada *Escenario* tendrá programada, en determinadas fechas, el comienzo de nuevas *listas de reproducción*. Las *zonas* junto con la

imagen o color de fondo formarán un *Escenario*. El detalle de la configuración de las *zonas* se puede consultar en el Anexo C. Manual del Usuario Administrador (UBICACIONES >> Configurar Ubicación >> Zonas a Esc). Se han definido cuatro *zonas* como máximo, ya que se ha llegado a la conclusión de que un número elevado de *zonas* simultáneas no centra la atención del posible espectador.

Listas (de reproducción).-. Una *Lista* contendrá una serie de archivos con ciertas características. El *Escenario*, junto con las *listas de reproducción* asociadas a cada *Zona* formarán una *Ubicación*; el detalle de la configuración de las *listas de reproducción* se puede consultar en el Anexo C. Manual del Usuario Administrador (UBICACIONES >> Configurar Ubicación >> Archivos a Listas).

Archivos.-. Se podrán subir *archivos* al servidor que posteriormente se programarán dentro de las *listas de reproducción*. Cada vez que asociemos un *archivo* a una *lista* se le podrán configurar ciertos parámetros que se tienen en cuenta cuando ese *archivo* se reproduce en una *Zona* concreta de un *Escenario*. El detalle de la administración de los *archivos* y la configuración de éstos en las listas de reproducción se puede consultar en el Anexo C. Manual del Usuario Administrador (ARCHIVOS >> Gestión de Archivos)

Público Real.-. Será el *público* que se encuentre delante de un elemento de proyección de contenido en un instante concreto. El *público real* es un concepto importante en el tipo de *Difusión Selectiva Adaptativa*, tal y como se ha definido en el apdo. 2.5.2, ya que el contenido emitido dependerá de la clasificación de este tipo de público en cada instante.

Público Objetivo o Público Potencial.-. Haremos uso de este término indistintamente cuando se haga referencia a los espectadores esperados en una determinada *Ubicación*. Por ejemplo, se podrían definir dentro del *Campus* de la *Universidad* dos *ubicaciones* diferentes: ‘*Cafetería*’ y ‘*Biblioteca*’. Se puede considerar como *público objetivo* diferente a estas dos ubicaciones aunque realmente sean las mismas personas el *público real* y vayan cambiando de *público objetivo* según se dispongan a comer o estén leyendo algún tratado. De este modo, en la *Ubicación* ‘cafetería’, se podría proyectar elementos publicitarios de menús o bebidas, mientras que en la *Ubicación* biblioteca se podrían proyectar las próximas conferencias o los últimos títulos adquiridos por la universidad. El *público objetivo* es un concepto clave ya que la *Difusión Selectiva* se determinará en la mayoría de los casos según sea este público y será determinante en los casos que se esté aplicando el tipo de *Difusión Selectiva Pasiva* tal y como se ha definido en el apdo. 2.5.1.

4.2.2 Definición de Elementos Específicos en la Arquitectura Desarrollada

El esquema general de la arquitectura final se muestra en la Ilustración 23; en ella se presentan los elementos que intervienen en el sistema de forma general:

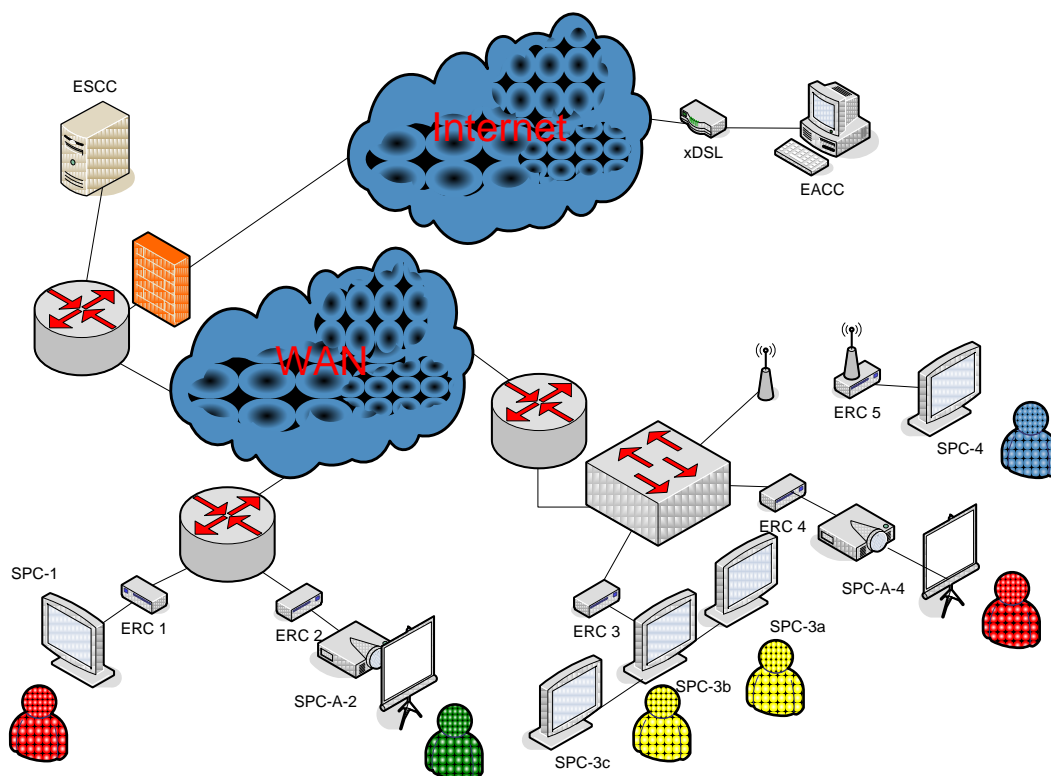


Ilustración 23.-. Arquitectura Definida

En el esquema se han representado todos los elementos importantes que intervienen en el sistema. El *Público Real* recibirá una programación diferente según esté observando una pantalla o *Sistema de Presentación de Contenido (SPC)* con una configuración u otra. A esta programación emitida se le ha denominado *Ubicación*, como hemos visto, y no se ha de confundir con el emplazamiento físico del *equipo terminal*.

Se presenta, a modo de tabla, un resumen de los elementos a considerar y que intervienen en el sistema implementado:

Icono	Definición	Descripción
	ESCC, Equipo Servidor de Contenido y Configuración.	Almacena la configuración de cada <i>Ubicación</i> . Proporciona el contenido multimedia según la configuración de cada ERC asociado a una <i>Ubicación</i> . La asociación a la <i>Ubicación</i> se realiza en función de la <i>dirección IPv4</i> del equipo y, según sea ésta, proporcionará un contenido u otro.
	Red WAN e Internet.	Representan las redes que conforman la conectividad entre los equipos. Se muestra dos tipos de redes: WAN e Internet. La WAN representa la red de área amplia, la cual conecta cada uno de los emplazamientos físicos con el servidor. Podría utilizarse incluso Internet para el transporte de la información entre sedes; no obstante, se hace referencia a una red WAN a lo largo del documento. Por otra parte, estableciendo las precauciones oportunas de seguridad, se puede realizar la configuración de las ubicaciones a través de una conexión desde Internet.
	Encaminador(Router)	Representan los <i>encaminadores</i> o <i>routers</i> que permiten establecer las conexiones a nivel de red. Estos elementos y las funcionalidades asociadas son realmente importantes cuando se aplica el sistema desarrollado en un <i>Espacio de Trabajo Distribuido</i> .

Icono	Definición	Descripción
	Conmutador (Switch)	Estos elementos ayudan a separar los dominios de difusión de las distintas <i>ubicaciones</i> además de proveer un punto de conexión física. Estos equipos, con una configuración apropiada, también permiten separar el tráfico de nuestra arquitectura o sistema (aplicación de <i>Difusión Selectiva</i>), del tráfico generado por otras aplicaciones que se encuentren, por ejemplo, en otra subred dentro de la misma red a través de VLANs.
	ERC, Equipo Reproductor Cliente	<p>Es un equipo <i>hardware</i> y que con ciertas aplicaciones <i>software</i> y configuraciones, se conecta de forma lógica al <i>ESCC</i> pidiendo que se le proporcione tanto el contenido multimedia, como la configuración asignada.</p> <p>El <i>ESCC</i> según la <i>dirección IPv4</i> del <i>ERC</i> le proporcionará una u otra configuración, ya que de esta forma se identificará en qué <i>Ubicación</i> se encuentra el <i>ERC</i> que realiza la petición.</p> <p>Los <i>ERC</i> estarán continuamente monitorizando si se ha cambiado la configuración de la <i>Ubicación</i> a la que pertenecen o se les ha cambiado de <i>Ubicación</i> a través de una conexión <i>socket</i> con el <i>ESCC</i>. El <i>ERC</i> será capaz de interpretar las órdenes que allí ‘escriba’ el <i>ESCC</i>.</p> <p>Estos equipos electrónicos estarán conectados directamente mediante conexiones físicas a los <i>Sistemas de Presentación de Contenido</i>.</p> <p>Los <i>ERCs</i> pueden ser de diferentes niveles de sofisticación y pueden ser configurados directamente o a través de servidores dependiendo de la clase de arquitectura implementada. En el contexto del proyecto se configurarán tanto de forma local, para obtener conectividad lógica, como desde el servidor para proveer la configuración que emitirá a través de una <i>arquitectura Cliente-Servidor</i>.</p>
	EACCC, Equipo de Administración, Configuración y Carga de Contenido	<p>Se denominan así, en el contexto de este proyecto, a los equipos informáticos utilizados para proveer la configuración al <i>ESCC</i> y a los <i>ERCs</i>.</p> <p>Desde estos equipos se puede cargar contenido multimedia al servidor <i>ESCC</i>, y el servidor distribuirá el contenido posteriormente a los diferentes <i>ERC</i> presentes en el sistema.</p>
	Conexión Inalámbrica	Simboliza la posibilidad de realizar la conexión a través de <i>Puntos de Acceso</i> (<i>APs</i> , <i>Access Point</i>) <i>Inalámbricos</i> en la infraestructura de red. Esta capacidad es importante cuando se quiere establecer equipos que mantengan cierta movilidad o que la provisión de estructura cableada no sea posible.
	SPC, Sistema de Presentación de Contenido	Representan dispositivos de presentación en forma de monitor (<i>TFTs</i> , <i>LCDs</i> , <i>CRTs</i> ,...) donde el <i>público objetivo</i> puede visualizar la <i>Difusión Selectiva</i> para ese emplazamiento físico.
	SPC-A, Sistema de Presentación de Contenido Alternativo.	Representa cualquier otro sistema de representación alternativo a pantallas o monitores. Se representa mediante un proyector y una pantalla aunque podría ser cualquier otro sistema avanzado: <i>video-wall</i> , <i>TV</i> ,...
	Ubicación i.	El icono representa que sobre el conjunto del <i>ERC</i> y el <i>SPC</i> o <i>SPC-A</i> se está aplicando la configuración para una <i>Ubicación i</i> concreta.

Tabla 11.-. Definición de elementos que intervienen en la Arquitectura.

Asumiendo estas definiciones se puede resumir la *arquitectura funcional* del *Macro* implementado.

La emisión de los archivos multimedia presentados en un *Sistema de Presentación de Contenido*, *SPC*, pueden ser controlados desde una conexión remota. La conexión se realiza desde un *Equipo de Administración de Configuración y Carga de Contenido*, *EACCC*,

(básicamente un dispositivo con conexión a la red) a un **Equipo Servidor de Configuración y Contenido, ESCC**, el cual tiene el rol de servidor y que envía la configuración y el contenido a todos los **Equipos Reproductores Clientes, ERCs**, que son los *equipos terminales* que reproducen el contenido multimedia mediante los *SPC*.

Se ha representado, en la Ilustración 23, el **Equipo Servidor de Contenido y Configuración (ESCC)** en una sede central y varios **Equipos Reproductores Clientes (ERCs)** repartidos geográficamente y conectados con el servidor a través de una *Red de Área Amplia (WAN)*, por lo que el esquema está representando un *Espacio de Trabajo Distribuido* como definiremos en apartados siguientes (apdo. 4.2.3 Espacios de Trabajo).

Se han representado, también, los elementos principales que intervienen en el sistema a nivel de red. Incluso se ha representando equipamiento electrónico que podría ser fundamental en un correcto diseño del sistema como es el **conmutador (Swicth)**. Según el fabricante y las capacidades del *Swicth* se puede separar en una misma red los equipos y las aplicaciones en diferentes subredes a través de **Redes Virtuales de Área Local (VLAN, Virtual LANs)**.

Otros elementos fundamentales dentro de la red son los **encaminadores (routers)**. Debe haber al menos uno en cada uno de los emplazamientos físicos diferentes que se han representado.

Los diferentes tipos de conexiones no forman parte de la arquitectura; es decir, el tipo de conexión no debe afectar a la *arquitectura funcional*. Se ha representado la conexión del equipo encargado de realizar la configuración del sistema, *EACCC*, a través de una conexión *xDSL* y un *PC* de uso doméstico, aunque podría ser cualquier, prácticamente, dispositivo con un navegador web y cualquier tipo de conexión que proporcione acceso al servidor.

También se han representado en el esquema, como se ha comentado, los **Equipos Reproductores Clientes (ERC)** conectados a diferentes **Sistemas de Presentación de Contenido (SPC)**: TFTs, LCDs, pantallas y proyectores. El *Equipo Reproductor Cliente (ERC)* será, básicamente, un sistema de *Hardware Embebido* o *Hardware Genérico* que realiza peticiones *http* al *ESCC* para obtener la configuración y que, además, es capaz de conectarse a un *socket* e interpretar las órdenes que desde allí se envíen.

El número de *ubicaciones* es, en principio, indefinido, aunque no ilimitado y dependerá en parte de las capacidades hardware del equipo servidor, la configuración establecida, el número de *ERCs* y el *ancho de banda* de las conexiones desde el *ESCC* hacia los *ERCs*.

4.2.3 Espacios de Trabajo

Para la *arquitectura funcional* del *Marco de Aplicaciones de Difusión Selectiva Diseñado y Desarrollado* se ha establecido una notación de los modos de funcionamiento, diferente a la utilizada en la *arquitectura de referencia*.

En la *arquitectura de referencia* se ha definido como *Espacio de Funcionamiento*, mientras que en la *arquitectura funcional implementada*, a este concepto similar, lo hemos denominado *Espacio de Trabajo*. Esto se ha decidido así debido a que, al contrario de lo que ocurre en la *arquitectura de referencia*, en nuestro sistema no se realizan cambios en las configuraciones por estar en un modo u otro. Tampoco es necesario incluir nuevos elementos en según qué *Espacio de Trabajo* estemos, por lo que el funcionamiento de la arquitectura en los diferentes modo es siempre igual. No obstante, como veremos en el Capítulo 8 Entorno de Pruebas y en

el Capítulo 9 Resultados de las Pruebas Realizadas , sí es posible evaluar las prestaciones atendiendo a estos diferentes *espacios de trabajo*.

La *arquitectura funcional* soporta dos de los *modos de funcionamiento* similares a los se han definido en la *arquitectura funcional de referencia*: *Espacio de Funcionamiento Simple* y *Espacio de Funcionamiento Distribuido*; los denominamos *Espacio de Trabajo simple* y *Espacio de Trabajo Distribuido*.

4.2.3.1 Arquitectura en Espacio de Trabajo Simple

Un ejemplo del esquema de la *arquitectura funcional* en *Espacio de Trabajo Simple* se muestra en la siguiente ilustración:

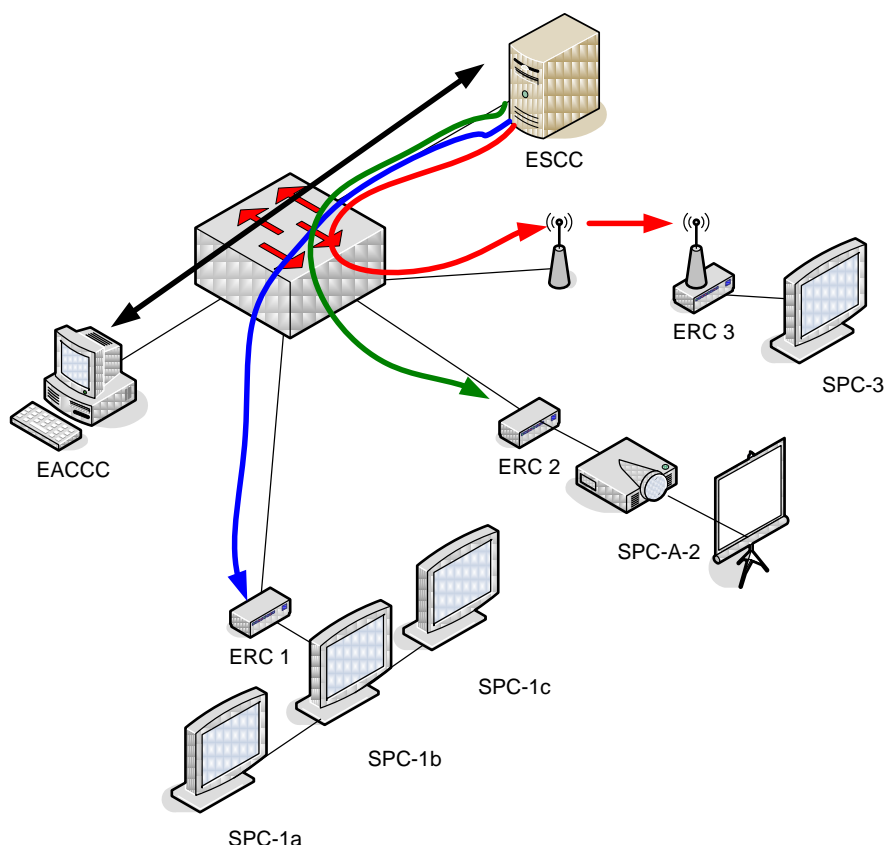


Ilustración 24.-. Ejemplo de Espacio Simple en la Arquitectura Funcional Implementada.

El aspecto fundamental y que marca la diferencia entre los distintos *espacios de trabajos* es que, en el tipo de *Espacio de Trabajo Simple*, no existe una *WAN* intermedia que limita el *ancho de banda* (de forma más restrictiva que una *LAN*) entre el *ESCC* y los distintos *ERCs* presentes en la solución. Como vemos en la figura anterior (Ilustración 24), se puede considerar un *Espacio de Trabajo Simple* cuando se tiene más de un *ERC*; sin embargo, la conexión entre *ESCC* y los *ERCs* es a través de una *LAN* y no una *WAN*.

El tipo de acceso a la configuración, del **Equipo de Administración de Configuración y Carga de Contenido (EACCC)** al *ESCC*, no condiciona el tipo de *espacio trabajo* en el que nos encontramos. Es posible mantener un *Espacio de Trabajo Simple* si la configuración y el contenido se gestionan a través de la *WAN* y la conexión desde **todos** los *ERCs* al *ESCC* se produce en una misma red y sin que haya una *WAN* intermedia en el camino de comunicación. Así, un *Espacio de Trabajo Simple* se muestra en la Ilustración 25, donde aparece una red *WAN*,

pero que sólo es utilizada para acceder a la configuración y subir el contenido multimedia al ESCC:

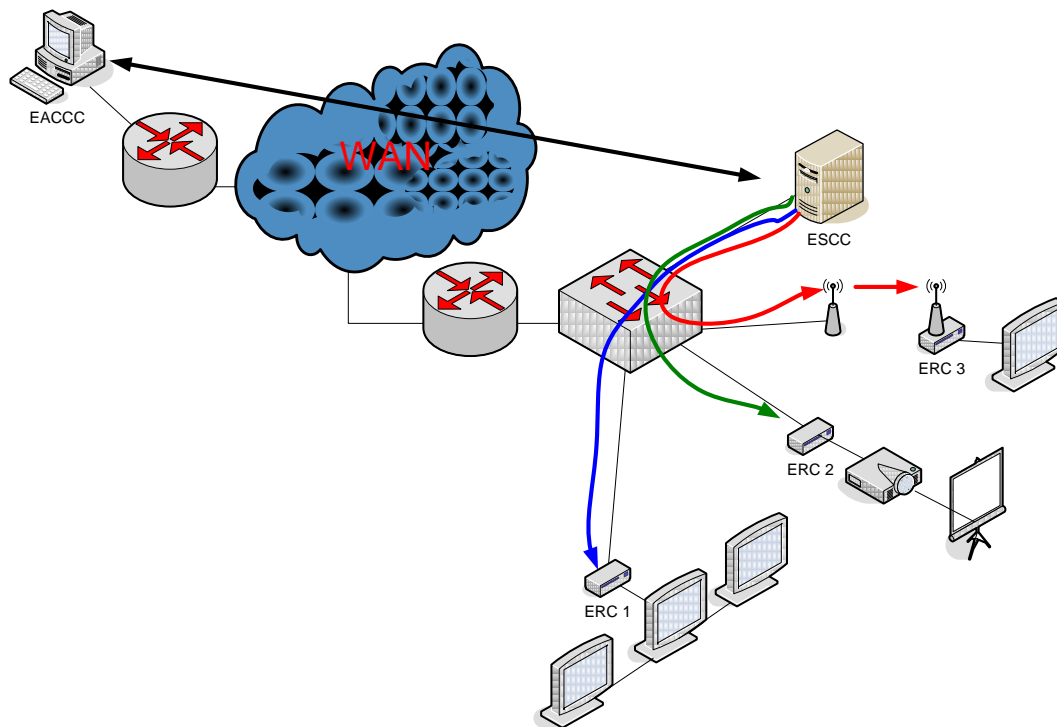


Ilustración 25.- Ejemplo de Espacio Simple donde aparece una conexión WAN para establecer la configuración y carga del contenido multimedia.

4.2.3.2 Arquitectura en Espacio de Trabajo Distribuido

Como se ha explicado, en un *Espacio de Trabajo Distribuido* hay una WAN que limita el *ancho de banda* a valores relativamente bajos. El *ancho de banda* de mayor interés es principalmente el enlace de bajada (*download*) desde el ESCC a los ERCs, ya que es el que envía el contenido y la configuración a los diferentes equipos.

A continuación (Ilustración 26) se muestra un ejemplo sencillo de un *Espacio de Trabajo Distribuido* dentro de la *arquitectura funcional implementada*:

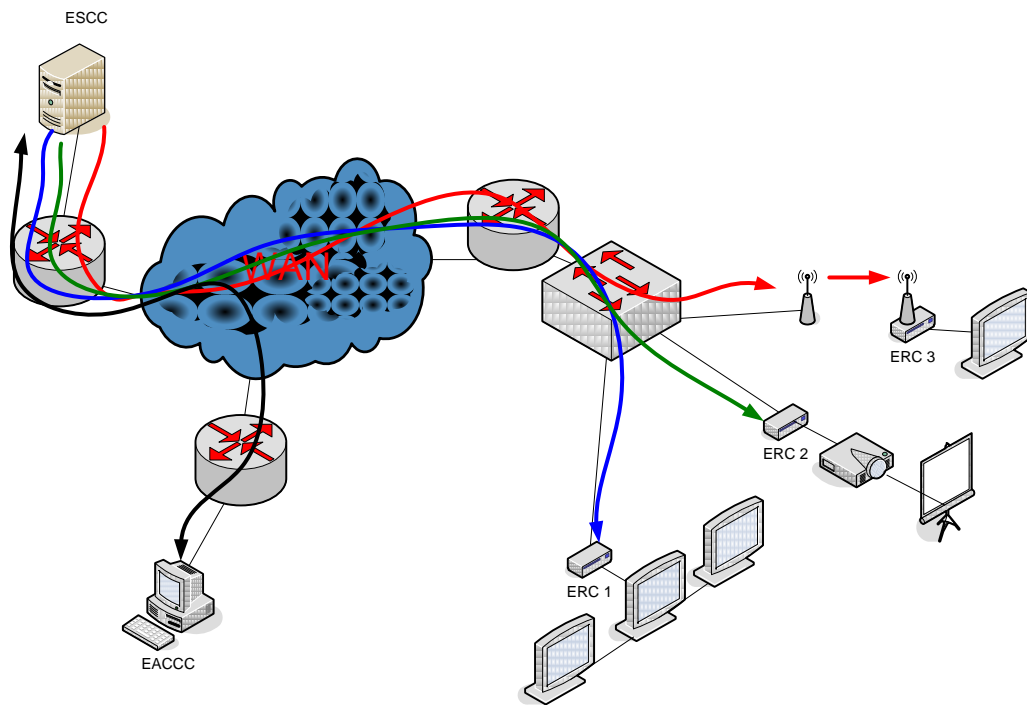


Ilustración 26.-. Ejemplo de Espacio Distribuido en la Arquitectura Funcional Implementada.

La diferencia determinante con respecto al *Espacio de Trabajo Simple* es que aparece una *WAN* entre la conexión del *ESCC* y los *ERCs*. Se ha de puntualizar que el hecho de que exista más de un *ERC*, no es indicativo de encontramos ante un *Espacio de Trabajo Distribuido* o en *Espacio de Trabajo Simple*.

A continuación (Ilustración 27) se muestra el esquema de un *Espacio de Trabajo Distribuido* con un solo *ERC*:

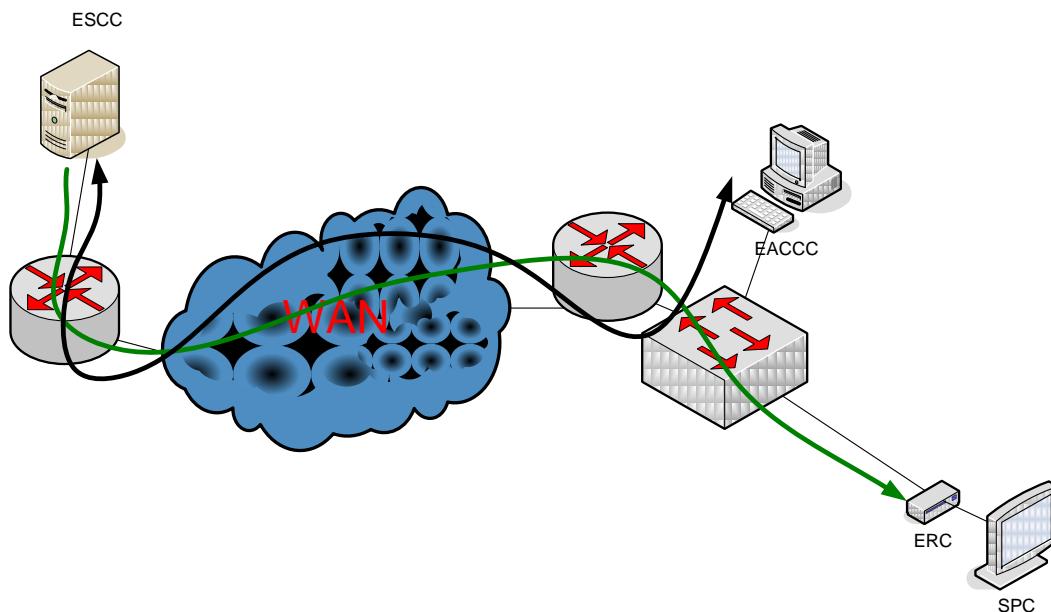


Ilustración 27.-. Ejemplo de Espacio Distribuido con un sólo ERC.

La Ilustración 27 muestra el *Sistema de Difusión Selectiva* desarrollado sobre una *arquitectura en Espacio de Trabajo Distribuido* en la que la configuración y la presentación de contenido se realizan desde una sede. Por tanto, se ha de recalcar que un *Espacio de Trabajo Distribuido* viene condicionado únicamente por la existencia de una *Red de Área Amplia (WAN)* entre el *ESCC* y los *ERCs*.

4.3 Arquitectura Funcional Definida para Aplicaciones de Difusión Selectiva

4.3.1 Funcionamiento General del Sistema Sobre la Arquitectura Funcional

En este apartado, se va a describir el funcionamiento general del sistema sobre la *arquitectura funcional*. De este modo en los siguientes capítulos, cuando se explique en detalle algunos elementos importantes del sistema de *Difusión Selectiva* (desarrollo y diseño del *Marco*), se obtendrá una visión completa del funcionamiento del sistema presentado.

En la Ilustración 28 se presenta un esquema que muestra el funcionamiento. En el esquema se supone un *Espacio Trabajo Simple* de la arquitectura; sin embargo, es extrapolable a una situación en la que la *arquitectura* corresponda con un *Espacio de Trabajo Distribuido*.

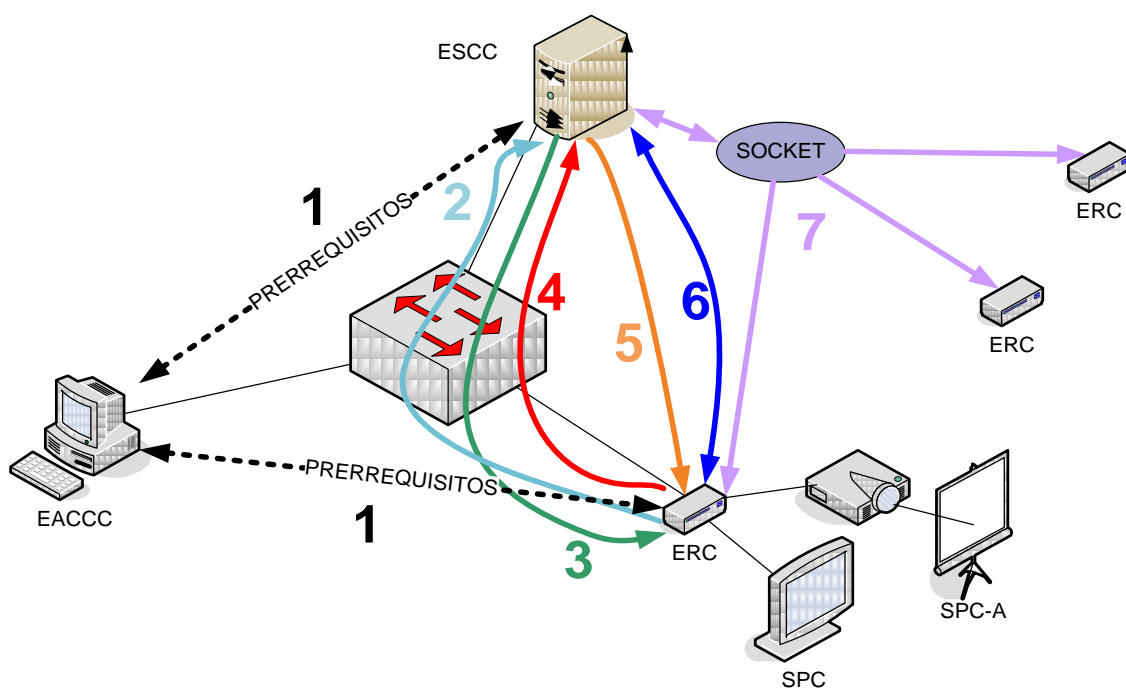


Ilustración 28.-. Resumen de Funcionamiento de la Arquitectura Funcional

Para resumir el funcionamiento general del sistema sobre la *arquitectura funcional* se detalla, en líneas generales, cada uno de los pasos presentados en la ilustración anterior, los cuales representan la secuencia de funcionamiento del sistema a través de los equipos que conforman la *arquitectura funcional*. Estos pasos, nos permiten construir una visión general del funcionamiento del sistema implementado:

1. Prerrequisitos:

- a) *Conexiones*: Se presupone que los equipos *ERCs* disponen de conectividad al *ESCC*, ya sea a través de una *WAN* (*Espacio de Trabajo*

Distribuido) o directamente en la misma red (*Espacio de Trabajo Simple*).

- b) *Puertos*: Se ha de permitir, en los *firewalls* presentes en el sistema, el acceso a varios puertos de los equipos. Esto es fundamental e indispensable para una correcta configuración y funcionamiento de los equipos (ver apdo. 4.6 Aspectos de Seguridad en las Conexiones), ya que se han habilitado servicios necesarios en algunos puertos de los equipos: *https*, *rsync*, *ntp* y otros.
- c) *VLAN y direccionamiento IPv4*: Se recomienda, dentro de una posible implantación de la *arquitectura del sistema de Difusión Selectiva desarrollado*, mantener el direccionamiento de la aplicación desarrollada en diferentes *VLANs* al del resto de aplicaciones. Así, el tráfico de las tramas dedicado a la *Difusión Selectiva* no afectaría al resto de aplicaciones del resto de la *LAN* y viceversa. Por otra parte, también se añade cierta seguridad al sistema; no obstante, realizando este tipo de aprovisionamiento a nivel de la *capa de enlace*, se ha de tener en cuenta que a nivel de la *capa de red* se deberá encaminar las *subredes IPv4* para que puedan alcanzar conectividad lógica a la *dirección IPv4* del *ESCC*.
- d) Configuración del *ESCC* (ver el apéndice C.I Manual del Usuario Administrador del *ESCC*).
- e) Configuración de los *ERCs* (ver el apéndice C.II Manual del Usuario Administrador del *ERC*).

2. *Petición de la Configuración (ERCs → ESCC).*

- a) Cuando los *ERCs* inician recuperan cuál es la *dirección IPv4* del *ESCC* al que están asociados. Este valor se almacena en la configuración de la *base de datos (MySQL)* particular de cada *ERC* (configurado en *Prerrequisitos*).
- b) Cada *ERC* pide la configuración vía la *dirección IPv4* recuperada en el punto anterior y realiza una petición *http* al *ESCC* en el puerto 8080. De este modo se realiza la petición para recuperar la configuración del *Escenario* de la *Ubicación* configurada para ese equipo.

3. *Envío de la Configuración del Escenario (ESCC → ERC).*

- a) El *ESCC* recibe la petición *http* sobre el puerto 8080, para identificar los valores de configuración de la respuesta, el servidor examina la *dirección IPv4* desde la cual se le está realizando la petición y decide a que *Ubicación* corresponde ese *ERC* (según se haya configurado en el propio *ESCC*).
- b) El *ESCC* envía la configuración del *Escenario* de la *Ubicación* que está configurada para esa *dirección IPv4* mediante *http/html*. Como decisión de diseño, si la *dirección IPv4* del *ERC* no está configurada en el *ESCC*, se envía la configuración de la *Ubicación 'defecto'*, la cual está presente siempre en la configuración, otra alternativa hubiera sido descartar la petición.

4. *Petición Configuración y Control (ERC → ESCC).*

- a) Al recibir la configuración del *Escenario* en *html*, se cargan cada uno de las aplicaciones necesarias (*swf*¹⁰) desde el *ESCC* en el *ERC*: la aplicación

¹⁰ *Swf (Small Web Format o ShockWave Flash)*: Es un formato de archivo de gráficos vectoriales creado por la empresa *Macromedia* (actualmente *Adobe Systems*); sin embargo, también admite *bitmaps* y vídeos. Necesita, para ser ejecutado, el *plugin Flash Player*, el cual permite mostrar las animaciones vectoriales que contienen los ficheros. Los archivos *swf* son compilados y comprimidos a partir de los

que reproduce los archivos de cada *Zona* y la aplicación de control de la *Ubicación* a la que está configurado el *ERC*.

- b) La *aplicación de reproducción* de la *Zona* realiza la petición, al *ESCC*, de las *listas de reproducción* de archivos configurados mediante nuevas peticiones *http* al puerto 8080.
- c) La *aplicación de control* de la *Ubicación* se conecta a un *socket* en el *ESCC* con el objetivo de interpretar las órdenes que allí se establecen.

5. Envío de la Configuración a cada Zona (ESCC → ERC).

- a) El *ESCC* responde a las peticiones *http* de la *aplicación de reproducción* de cada *Zona* del *ERC* en código *XML*. Este código *XML* muestra el contenido de la *lista de reproducción* (ordenando los archivos por los intervalos de reproducción) y la fecha a partir de la cual se debe cambiar de *lista de reproducción*.

6. Petición/Recepción del contenido Multimedia mediante Pseudo Streaming (HTTP Streaming) (ESCC ↔ ERC)

- a) Los *ERCs*, al recibir la lista de archivos a reproducir, realizan un procesamiento de los intervalos de la lista de reproducción. Al procesar los intervalos se activa un temporizador que, al cumplirse, lanza un evento que hace que comience a reproducir los archivos del siguiente intervalo.
- b) Los *ERCs*, al recibir también la fecha de *lista de reproducción* siguiente en la respuesta *XML* del *ESCC*, activan un segundo temporizador de modo que, al vencer éste, se produce un evento que hace que se vuelva a realizar la petición al *ESCC* para obtener la nueva *Lista*.
- c) El procesamiento de estas respuestas se realiza en la propia *Zona* de presentación y reproducción del contenido mediante una *aplicación swf* (*aplicación reproducción de Zona*). Cada *Zona* ejecuta la aplicación de forma independiente, debido a que la configuración de las *listas de reproducción* se asigna a cada *Zona* de la *Ubicación*.
- d) Según sea el estado de sincronismo, entre el *ESCC* y el *ERC*, relativo al contenido multimedia almacenado de forma local en el *ERC*, se realiza la petición de contenido multimedia (*HTTP Streaming*) al *ESCC* o de forma local. Es decir, se ha desarrollado un mecanismo *Auto-Proxy* (ver apdo. 4.3.2 Arquitectura Funcional Auto-Proxy).

7. Procesado de la Información Socket.

- a) El *ERC* se conecta mediante la *aplicación de control* de la *Ubicación* a un *servidor socket* del *ESCC* en puerto *TCP 8081*. Este *servidor socket*, que es el encargado de distribuir información a los *ERCs*, envían las órdenes al resto de equipos que están escuchando. Esta información, que escribe el *ESCC*, se intercambia mediante formato en código *XML*.
- b) La *aplicación de control* conoce a qué *Ubicación* corresponde a través de parámetros que se pasan a la aplicación al recibir la configuración del *Escenario* (paso 3). Así, cuando el *ESCC* escribe en el *socket*, el *ERC* sabe si debe interpretar la orden debido a que conoce a qué *Ubicación* pertenece.
- c) Una vez que el *ERC* ha identificado una orden, dirigida a él toma la acción correspondiente: actualizar la configuración con nuevas peticiones al *ESCC* o sincronizar contenido con el *ESCC*

archivos editables (en formato *.fla*) con los se trabaja en *Adobe Flash*, o escritos en el formato *ASC (.sc)*, basado en texto, que pueden ser escritos y editados en cualquier editor de texto y compilados por el *SWFC* (*compilador swf*).

Con este funcionamiento se tiene un sistema de *Difusión Selectiva*, ya que la *lista de reproducción* obtenida en cada *Zona*, de cada *Escenario* y, por tanto, en cada *Ubicación*, depende de la *dirección IPv4* del equipo cliente que realiza la petición.

4.3.2 *Arquitectura Funcional Auto-Proxy*

Un condicionante adicional a la hora de elegir una arquitectura *Cliente-Servidor* es que a través de una arquitectura de este tipo es más sencillo modificarla para introducir equipos intermedios que funcionen como *proxy*, tanto en la atención de las peticiones, como la provisión del contenido.

Dentro del contexto del proyecto se define como *proxy* a un elemento de la red que es capaz de proveer las necesidades para aprovisionar el contenido multimedia (archivos) a los diferentes *ERCs*. Es decir, los archivos de video e imágenes se recuperan directamente desde el equipo *proxy* según se ha implementado el sistema. Por tanto, introducir un elemento intermedio, *proxy*, en el sistema tiene, como fin último, intentar minimizar el consumo del *ancho de banda* de la *WAN* dentro de la arquitectura y en última instancia hacer lo propio con el *ancho de banda* de la *LAN* si fuera necesario.

Según se ha indicado, el *ERC* conoce el contenido que tiene que reproducir a través de la *Lista* que le ha proporcionado el *ESCC* en formato *XML* (paso 6 del apdo. 4.3.1 Funcionamiento General del Sistema Sobre la Arquitectura Funcional). Con esta información, el *ERC* identifica si tiene almacenado el vídeo de forma local para su reproducción, aplicando así un mecanismo al que hemos denominado *Auto-Proxy*. Si fuera así, y el contenido estuviera presente de forma local en el *ERC*, el archivo se reproduciría mediante *HTTP Streaming* a *localhost* y, en caso contrario, se realizaría la petición al *ESCC* para establecer el *HTTP Streaming* desde la aplicación encargada de reproducir el contenido de la *Zona*.

Cada *ERC* funcionará como su propio equipo *proxy*, de modo que debe sincronizar los archivos de forma local con los almacenados en el *ESCC*. Con este mecanismo (*Auto-Proxy*) aparentemente se utiliza el *ancho de banda* de forma poco eficiente, ya que cuando surge algún cambio en los archivos disponibles en el *ESCC*, hay que actualizar el contenido desde el *ESCC* a todos los *ERCs*. Sin embargo, con este mecanismo la eficiencia en la *WAN* y en la *LAN* es completa cuando el sistema se mantiene funcionando de forma estable y que será la mayor parte del tiempo en una aplicación de *Cartelería Digital*.

En el siguiente esquema (Ilustración 29) se presenta la evolución desarrollada de la arquitectura introduciendo elementos *Auto-Proxy* en cada uno de los *ERCs*.

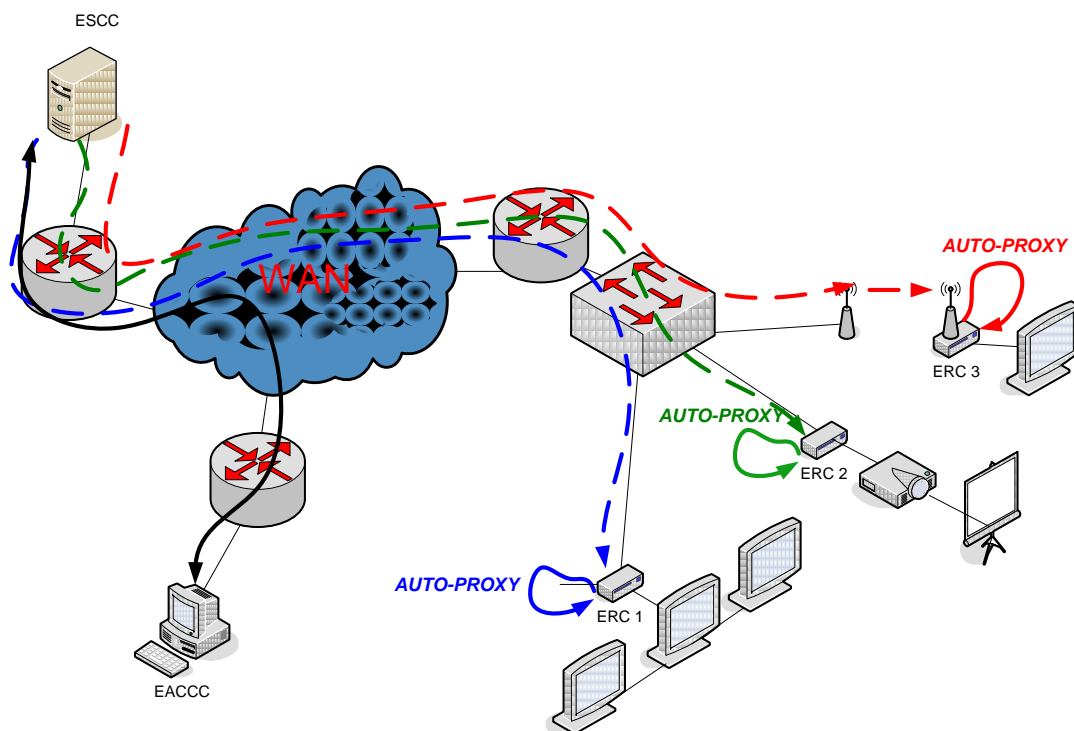


Ilustración 29.-. Arquitectura Funcional con característica Auto-Proxy en cada uno de los ERCs.

En esta arquitectura los *ERCs* realizan una copia del contenido almacenado inicialmente en el *ESCC*, acción que viene representada por las líneas discontinuas en la Ilustración 29. De este modo, se produce una sola copia por cada *ERC* de todo el contenido multimedia. El ancho de banda *WAN* queda con una mayor eficiencia en parte, ya que para el caso de esta arquitectura, el ancho de banda *WAN* queda ocupado al inicio de la sincronización¹¹. De esta manera, en reproducciones sucesivas del mismo contenido, no se utilizará la *WAN* para obtener el contenido, ya que los *ERCs* lo tendrán almacenado de forma local y tan sólo se realizarán peticiones de configuración, las cuales utilizan un pequeño ancho de banda.

Este tipo de arquitectura es altamente recomendable en situaciones donde el contenido multimedia se mantiene estático en largos periodos de tiempo (días) de modo que no se está continuamente modificando los archivos almacenados en el *ESCC*, y es una importante mejora que se ha introducido dentro de la arquitectura inicial.

Existen otros muchos posibles desarrollos de arquitecturas similares. Es posible adquirir el contenido de forma local sólo para los archivos presentes en la configuración de la *Ubicación* a la que pertenece el *ERC*; no obstante, aunque se obtiene una mayor eficiencia del *ancho de banda*, es posible que el funcionamiento no mejore en determinadas circunstancias. Por ejemplo, si a un equipo se le cambia de *Ubicación* pediría al *ESCC* la sincronización de todos los archivos que intervienen en la configuración de una determinada *Ubicación* (si no coincidieran con los de la *Ubicación* anterior). Todos los archivos podrían generar un flujo importante y durante un periodo de tiempo considerable debido a la cantidad y tamaño de los archivos multimedia. Si esta situación se diera en una cantidad importante de *ERCs* (muchos equipos configurados a la misma nueva *Ubicación*) la situación empeoraría; sin embargo, con la

¹¹ Finalmente se incluyó, como veremos, la restricción de uso de *ancho de banda* por cada *ERC* para evitar sobrecargar la red *WAN*. La restricción del *ancho de banda* permitido para la sincronización es un parámetro de configuración de cada *ERC*.

implementación establecida (descargar de forma completa todos los archivos de forma local), un cambio de *Ubicación* de un determinado equipo (o varios) no afectará al sistema en temas relacionados con el *ancho de banda* consumido por los *ERCs* implicados en la nueva configuración.

Como veremos en el anexo C.II Manual del Usuario Administrador del ERC, una forma que ha sido utilizada y desarrollada en el proyecto, para la solución finalmente adoptada, de relajar la situación y de no sobrecargar la red en términos de *ancho de banda* utilizado, es limitar el *ancho de banda* que cada *ERC* puede utilizar para la sincronización de archivos con el *ESCC*. Este parámetro es parte de la configuración de los *ERCs*.

4.4 Alternativas de Arquitectura Funcional Valoradas

Existen muchas alternativas en el desarrollo de la *arquitectura funcional*. Así, partiendo de la definición de *proxy* que hemos dado (apdo. 4.3.2 Arquitectura Funcional Auto-Proxy) se plantean y definen otras posibilidades para implementarla:

- **Arquitectura Proxy-Caché:** Con esta arquitectura se plantea mejorar el uso del *ancho de banda* de la *WAN*, de modo que un solo equipo de cada subred o emplazamiento remoto se descarga el contenido multimedia y lo aprovisiona al resto de equipos de la *LAN*. La eficiencia se consigue en la *WAN*, estando la *LAN* llena de peticiones hacia el equipo *Proxy-Caché* desde el resto de los *ERCs* del emplazamiento.
- **Arquitectura Proxy-Caché/Auto-Proxy con provisión Distribuida (P2P):** Esta arquitectura sería funcionalmente similar a las presentadas; es decir, se establece cualquiera de los mecanismos *proxy* a la hora de reproducir el contenido. No obstante, el contenido multimedia se distribuye atendiendo a técnicas *P2P* [70] [71] [72]. Así, cada elemento que forma parte de la red *P2P* es un nodo que comparte el contenido con el resto de nodos.

En el desarrollo de estas implementaciones se ha de tener en cuenta que la programación de la *lista de reproducción* de cada *Ubicación* se solicita al *ESCC* y que el uso eficiente del *ancho de banda* se realiza sobre los elementos que generan un tráfico significativo que son los involucrados en el envío y recepción de archivos multimedia y no las peticiones de configuración.

4.4.1 Arquitectura Proxy-Caché

Este apartado presenta un esquema de la arquitectura con elementos intermedios *Proxy-Caché*, el cual se muestra como simple ejemplo y puede ser considerado su desarrollo y estudio de funcionamiento como línea de trabajo futuro. No obstante, la presentación de esta arquitectura es útil para compararla con la finalmente desarrollada.

La Ilustración 30 representa el funcionamiento de la arquitectura *Proxy-Caché*.

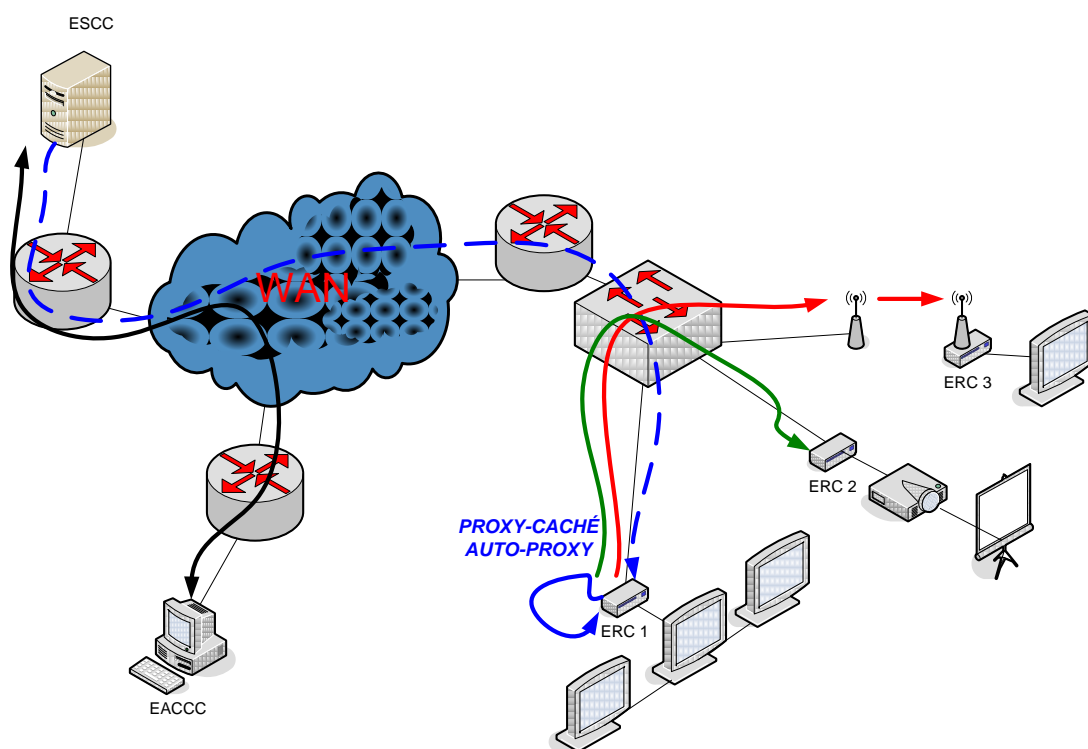


Ilustración 30.-. Arquitectura Funcional con elementos Proxy-Caché.

En la ilustración anterior, se muestra cómo quedaría la arquitectura que se ha desarrollado con un ERC con rol de *Proxy-Caché*, de forma similar al que se utiliza en la *Arquitectura de Referencia* en el que hemos denominado *Espacio de Funcionamiento Distribuido* (ver Apdo. 3.4.3.3). La línea discontinua indica que actualiza y sincroniza el contenido de los archivos con el del servidor principal, el ESCC, de modo que recupera el contenido necesario para las distintas *Ubicaciones*. Así el ancho de banda WAN se utiliza una sola vez por archivo y emplazamiento y no cada vez que se reproduce el archivo en cada ERC de ese emplazamiento.

De esta manera las peticiones para la reproducción de los archivos se realiza desde los ERCs hacia al equipo *Proxy-Caché* cuyo flujo se representa en línea continua. En esta posible modificación de la arquitectura se maximiza en eficiencia el *ancho de banda* utilizado de la WAN y se aproxima mucho a la arquitectura de referencia en *Espacio de Funcionamiento Distribuido* (ver apdo. 3.4.3.3 *Espacio de Funcionamiento Distribuido*).

Cabe la posibilidad de desarrollar una *arquitectura híbrida* entre la arquitectura *Proxy-Caché* y la arquitectura *Auto-Proxy*. Esta alternativa consistiría en mantener por una parte un ERC como elemento *Proxy-Caché* y que el resto de ERCs de la red sincronizaran el contenido como *Auto-Proxy* con este elemento *Proxy-Caché* en vez de directamente con el ESCC; incluso cabría la posibilidad de establecer una arquitectura híbrida en árbol. Esta forma de funcionamiento de la arquitectura, *arquitectura híbrida*, se deja también como línea de trabajo futuro (ver apdo. 10.2 Líneas de Trabajo Futuro) para su implementación y pruebas de funcionamiento, debido a que el sistema, en una situación estable de funcionamiento, no mejora el consumo de *ancho de banda* con respecto a la arquitectura finalmente desarrollada aunque sí lo hace en estado de sincronización de los ERCs. Como contrapartida, el tiempo de provisionar a todos los ERCs con el archivo almacenado de forma local podría aumentar en determinadas situaciones y el funcionamiento de la arquitectura sería más complejo.

4.4.2 Arquitectura Distribuida para Provisión de Contenido: P2P

Según se ha establecido en el *Marco* finalmente implementado, los archivos se descargan de forma local en el *ERC* a través de una aplicación que sincroniza directamente los archivos de ciertas carpetas del *ESCC*. Con el objetivo de intentar optimizar el uso del ancho de banda en las diferentes conexiones se plantea la posibilidad de implementar algún mecanismo *P2P* (*Peer-to-Peer*):

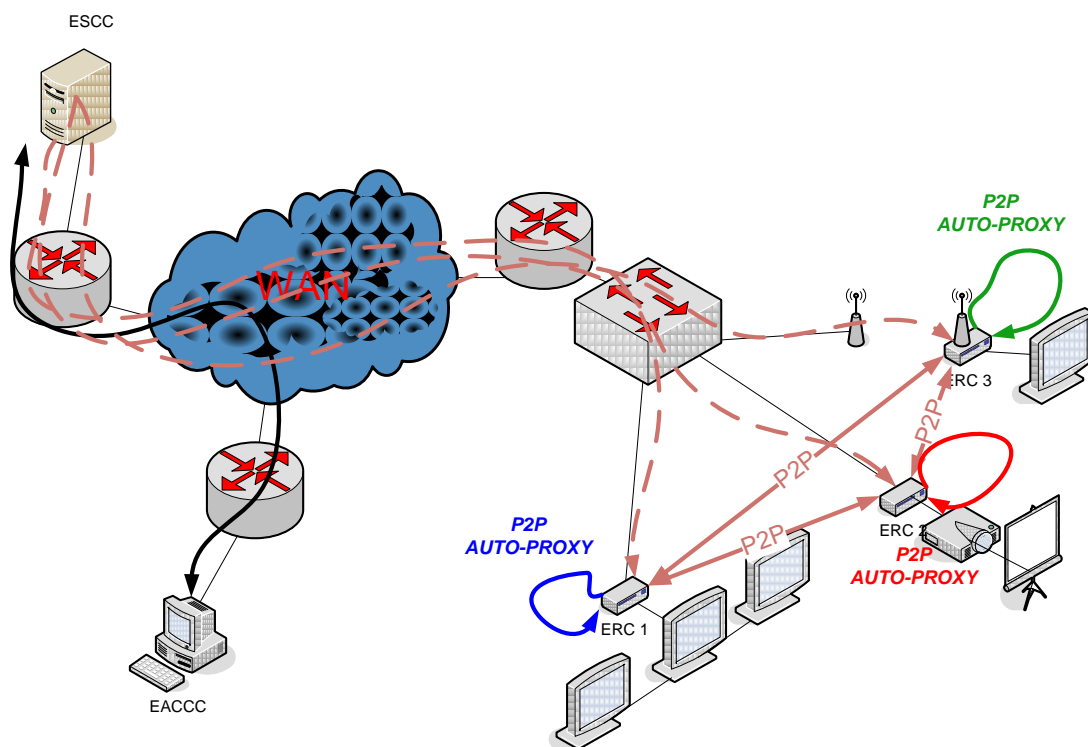


Ilustración 31.-. Arquitectura Funcional con elementos Auto-Proxy y distribución de contenido utilizando técnicas P2P.

Se ha de tener en cuenta que la utilización de redes *P2P* debe permitir el direccionamiento IP entre las diferentes sedes, mientras que si se actualiza el contenido directamente desde el servidor, cada *ERC* sólo necesita poder comunicarse con éste.

También cabe destacar que en sistemas de este tipo tiene importancia el tipo de conexión que se establezca en las sedes remotas, sobre todo en conexiones asimétricas entre sedes.

4.5 Sincronización de la Hora en la Arquitectura Funcional

Un elemento muy importante que se observó al desarrollar el *Marco* es que dentro de la *arquitectura funcional* el *ERC* debe establecer la misma hora que el *ESCC*.

Cada vez que el *ERC* se conecta al *ESCC* realizando una petición de conexión, recibe la configuración para la *Ubicación* al que el equipo está asignado, la cual se la proporciona el *ESCC* según la configuración que el administrador haya establecido.

El *ESCC* envía la configuración de la *Ubicación* dependiendo de la fecha local configurada en el propio equipo indicando los archivos que debe emitir la *lista de reproducción* actual en

un determinado horario e indicando la fecha de la próxima *lista de reproducción* de la *zona* de la *Ubicación* concreta.

Por otra parte, el propio *ERC* decide cuándo debe realizar peticiones de configuración al *ESCC* a través del cálculo de dos temporizadores: uno para los intervalos dentro de la configuración de la *lista de reproducción* y el otro cada vez que se vence la fecha para la siguiente *lista de reproducción*. Estos temporizadores se establecen a partir de la hora local que tiene almacenada en cada *ERC*.

A continuación, mediante un ejemplo genérico y de forma ilustrativa, se presenta la funcionalidad desarrollada en la aplicación que ilustra la necesidad de incorporar un mecanismo de sincronización de hora:

- El sistema *ERC* establece, a partir de los datos de configuración obtenidos del *ESCC* relativos a la programación de la *Ubicación* y del propio reloj (el cual tiene sincronizado con el *ESCC*), la siguiente información:
 - “Ahora son las $\langle h_1h_1:m_1m_1:s_1s_1 \rangle$, del día D_1D_1 del mes M_1M_1 del Año $A_1A_1A_1A_1$ ”
 - “El siguiente intervalo de reproducción de archivos de la lista que se está reproduciendo comienza a las $\langle h_2h_2:m_2m_2:s_2s_2 \rangle$ de hoy.”
 - “la siguiente lista de reproducción comienza el día D_2D_2 de mes M_2M_2 del Año $A_2A_2A_2A_2$ ”
- Por tanto, se han realizado varios cálculos basados en la sincronización de tiempos entre equipos:
 - “faltan X_1 segundos para el siguiente intervalo de reproducción ($\langle h_2h_2:m_2m_2:s_2s_2 \rangle$) de archivos de la lista en reproducción.”
 - “faltan X_2 segundos para el día siguiente; es decir para el día D_3D_3 de mes M_3M_3 del Año $A_3A_3A_3A_3$ ”

- Si el día siguiente ($\langle D_3D_3-M_3M_3-A_3A_3A_3A_3 \rangle$) es igual, al día de la siguiente fecha de reproducción ($\langle D_2D_2-M_2M_2-A_2A_2A_2A_2 \rangle$):

$$\langle D_3D_3-M_3M_3-A_3A_3A_3A_3 \rangle == \langle D_2D_2-M_2M_2-A_2A_2A_2A_2 \rangle$$

Activamos un *flag* que indica realizar petición a la configuración de archivos al servidor por cambio de lista de lista de reproducción.

- Si no:

Desactivamos *flag* que indica realizar petición a la configuración de archivos al servidor por cambio de lista de reproducción.

- Se registran dos eventos en la aplicación que controla el estado del *ERC* en particular:
 - ENVENTO T_1 : Vence el temporizador X_1 :
 - Realizamos una petición al servidor *ESCC* con la configuración del siguiente intervalo de archivos de reproducción para *zona* de la *Ubicación*.
 - ENVENTO T_2 : Vence el temporizador X_2 :

- Realizamos una petición al servidor *ESCC* de la nueva *lista de reproducción* y los intervalos para la *Zona* de la *Ubicación*

Por tanto, si nos fijamos en detalle en el mecanismo funcional diseñado, es evidente la importancia del hecho de que ambos roles de equipos (*ERC* y *ESCC*) mantengan la hora sincronizada, ya que el *ERC* determina, según la fecha y hora que maneja, cuándo tiene que realizar peticiones al *ESCC* y, por otra parte, la configuración a enviar se decide en el *ESCC* en función de la hora y fecha que tiene éste configurado.

Teniendo esto en cuenta, se ha de indicar que se puede resolver esta necesidad de sincronización temporal con varios métodos. Por ejemplo, en cada respuesta del *ESCC*, éste podría enviar la fecha y hora que tiene configurado a los equipos *ERCs* de modo que estos calculen los temporizadores a partir de éste valor. Otra alternativa, aunque siendo ésta una peor opción, es que el *ERC* envíe en la petición la hora y fecha y el *ESCC* envíe la configuración como si nos encontrásemos en ese instante. Como se comentaba esta segunda opción es peor solución, porque la emisión de contenido dependería en cada *ERC* de la hora que estuviera configurada en el equipo no estando la hora del sistema sincronizada o referenciada en un único sistema. Este hecho podría provocar situaciones en las que equipos *ERC* pertenecientes a la misma *Ubicación* estuvieran emitiendo contenido diferente en el mismo instante, situación indeseable.

La solución finalmente adoptada en nuestro sistema ha sido utilizar el *protocolo NTP* para la sincronización de la hora de los equipos. Así, se ha establecido el servidor *ESCC* como *referencia NTP* para el resto de equipos *ERCs*.

No se ha establecido una interfaz web para modificar la hora del *ESCC*, simplemente se usan los servidores de hora *NTP* como fuente del reloj del sistema. La configuración de estos servidores *NTP* se realizará a través del interfaz web desde el equipo administrador (ver C.I Manual del Usuario Administrador del *ESCC* (SISTEMA >> Configuración Servidor >> Referencias NTP)).

4.6 Aspectos de Seguridad en las Conexiones

Será decisión del propio usuario establecer las políticas de seguridad dentro de la red. Desde el punto de vista de diseño, se ha establecido un sistema a partir del cual con un navegador web cliente los usuarios tendrán acceso a la aplicación que configura los diferentes *Ubicaciones: Escenarios* y las *listas de programación* que se proyectan en cada una de las *Zonas*. Por tanto, se establece que esta conexión de configuración se debería realizar a través de una comunicación segura a través del servidor web en el que descansa la aplicación.

Se ha realizado un diseño adoptando ciertas políticas de seguridad básicas. A continuación se enumeran de forma resumida algunos criterios que se deberían adoptar en la solución y que se considerarán como “*Mejores Prácticas de Diseño*”:

- **VLAN Específica:** Se espera que en el diseño de la red de acceso al medio, el tráfico del sistema quede de forma independiente del resto de tráfico de red. A efectos prácticos, la red de *Difusión Selectiva* debe quedar en una subred diferente del resto de aplicaciones de la *LAN*. Además de proveer cierta seguridad también permite separar dominios de difusión.
- **Conexiones VPN:** Las conexiones cifradas desde los equipos *ERCs* al *ESCC* pueden llegar a ser un elemento importante en la seguridad del sistema, principalmente por aspectos relacionados con la confidencialidad. Se espera que en la configuración de la

arquitectura, al menos entre los *routers*, se establezcan *túneles virtuales seguros* para que así el flujo de datos se mantenga seguro si se hace uso de redes públicas.

- **Seguridad Lógica:** Los *firewalls* y *routers* podrían ser configurados con listas de acceso, además de permitir únicamente los puertos implicados en las comunicaciones de la arquitectura. Se han definido los siguientes puertos en el *Marco* finalmente desarrollado: 443 (*https*), 8080 (*http*), 123 (*ntp*), 843 (*permisos socket*), 8081 (*socket*) y *rsync* (873). Por tanto, se ha de tener en cuenta abrir los diferentes puertos en los *firewalls* que estén presentes en la red:

- (1) Puertos TCP desde peticiones de los ERCs hacia el ESCC: peticiones *http* (8080), *servidor socket* (8081), *permisos socket* (843), *rsync* (873), *ntp*¹² (123). Es decir, se debe permitir a los ERCs acceder a estos puertos del ESCC.
- (2) Puertos TCP desde peticiones del EACCC hacia el ERC y ESCC: administración web *https* (443), *servidor socket* (8081, sólo ESCC), *permisos socket* (843, sólo ESCC), *aplicación ssh* (22). Estos puertos permiten al administrador del sistema y tener el control para la administración del *Marco*.

Estos aspectos de seguridad de las conexiones se deberían establecer para el funcionamiento en producción de la arquitectura desarrollada. Sin embargo, más allá de los aspectos de seguridad, el funcionamiento correcto del sistema no se vería afectado por esta configuración.

4.7 Resumen

En el capítulo hemos definido una *arquitectura funcional* partiendo de unos requisitos que se han considerado que deberían cumplir cualquier *arquitectura funcional* para ofrecer una base para el desarrollo de un sistema de *difusión selectiva*. En el contexto del proyecto fin de carrera la *arquitectura funcional* será la base sobre la que se diseñará el *Marco* para ejecutar aplicaciones de difusión selectiva de contenido multimedia.

En este capítulo se ha realizado la definición de la terminología específica para el sistema finalmente desarrollado (*Ubicación, Escenario, Zona, Lista, Archivo, Público Real, Público Objetivo*) y que será utilizada a lo largo de los capítulos para explicar el trabajo realizado y, así, eliminar las posibles ambigüedades. También, con el mismo espíritu, se han definido explícitamente los diferentes elementos que componen la *arquitectura funcional* (ESCC, ERC, EACCC, SPC, SPC-A) y se han matizado las funciones de los dispositivos de red que son necesarios para generar la *arquitectura funcional* (Red WAN, Router, Switch, Conexión Inalámbrica).

Se ha realizado una distinción acerca de los *espacios de trabajo* considerados en el proyecto fin de carrera y las diferencias existentes con los *espacios de funcionamiento* definidos para explicar la *arquitectura de referencia*. La distinción entre el *Espacio de Trabajo Simple* y el *Espacio de Trabajo Distribuido* se establece por la ausencia o presencia, respectivamente, de una WAN entre el ESCC y los ERCs. En el sistema de referencia los *espacios de funcionamiento* modifican la *arquitectura funcional* allí establecida, introduciendo nuevos equipos con diferentes roles, ante la existencia de una WAN. La *arquitectura funcional implementada* se mantiene invariante en todos los *espacios de trabajos*, con independencia de si se introducen una WAN entre el ESCC y alguno de los ERCs.

¹² Para la configuración del protocolo *NTP* también se ha de habilitar el puerto *UDP* 123.

En el capítulo, también se ha presentado el funcionamiento del sistema de difusión selectiva sobre la *arquitectura funcional* y se ha subrayado que la definición de ésta influirá en el diseño y desarrollo del *Marco* y en la interacción de los elementos que lo formen. Se ha definido una *arquitectura funcional* a la que hemos denominado *Auto-Proxy*, la cual permitirá realizar un uso eficiente del *ancho de banda*. La eficiencia se consigue estableciendo *HTTP Streaming* de forma local en los *ERCs* previa descarga del contenido multimedia. El uso eficiente del *ancho de banda* se hace necesario en una configuración de *Espacio de Trabajo Distribuido* donde los *ERCs* se conectan al *ESCC* a través de una *WAN*.

En este capítulo, también se ha mostrado la necesidad de establecer algún mecanismo de sincronización de hora entre los *ERCs* y el *ESCC* y se han enumerado algunas consideraciones de seguridad que se deberían tener en cuenta en la configuración de una red que ejecute el sistema de difusión selectiva sobre la *arquitectura funcional definida*.

Capítulo 5 Descripción del Marco Diseñado y Desarrollado

En el presente capítulo se explica el *Marco* desarrollado (*software* desarrollado e integración realizada) para alcanzar la *arquitectura funcional* mostrada en el Capítulo 4.

El *Marco* debe proporcionar las herramientas para:

- Proveer la configuración y programación del contenido multimedia de los distintos *ERCs* a través de una *Interfaz Gráfica de Usuario* administrador del sistema que permite acceder y configurar el *ESCC*.
- Proveer la configuración a los *ERCs* para que permitan alcanzar la conectividad con el *ESCC*.
- Permitir a los *ERCs* acceder a la configuración de las *listas de reproducción* programadas en el *ESCC* y mostrar el contenido multimedia de acuerdo a la programación.

Se enumerarán algunas características que deben cumplir el diseño y las funcionalidades iniciales del *Marco*. A partir de esas características se han valorado las diferentes opciones de las tecnologías disponibles para implementar la *arquitectura funcional* del sistema mediante el desarrollo de un *Marco*, seleccionando las que mejor se adaptan a nuestras necesidades.

En este capítulo también se explica cómo se han conseguido cada uno de los objetivos con las tecnologías *software* elegidas, detallando el mecanismo lógico de funcionamiento de los elementos clave y generando un *Marco para Aplicaciones de Difusión Selectiva*.

Por lo anterior, éste capítulo se divide en varios grandes bloques:

1. Se justifica la elección de la tecnología para la implementación del *Marco* haciendo un breve resumen de las principales aplicaciones utilizadas: *Apache*, *MySQL*, *php* y *ActionScript3*. Se explica el razonamiento para la elección de éstas, analizando brevemente alguna de las alternativas.
2. Se describe la implementación del *Marco* que se ha diseñado para lograr el objetivo de obtener un sistema de *Difusión Selectiva*.
3. Se resume el *Entorno de Desarrollo* configurado.
4. Se muestran las configuraciones realizadas y la lógica de los elementos principales desarrollados para obtener el *Marco* para aplicaciones de *Difusión Selectiva* diseñado. Las explicaciones de las configuraciones y los desarrollos se han dividido en apartados donde las configuraciones son comunes a ambos equipos (*ERC*, *ESCC*) y se ha descrito, por otra parte, las configuraciones y los desarrollos específicos de cada equipo principal definido en la *arquitectura funcional*.

5.1 Elección de la Tecnología Software

La elección de la tecnología es una decisión importante dentro del *Marco* desarrollado ya que ésta condicionará la forma de conseguir los diferentes objetivos. Así, para una correcta elección se han de definido inicialmente las necesidades del sistema final, tanto a nivel de usabilidad por parte del administrador, como de características de la propia arquitectura funcional, las cuales se han enunciado en el Capítulo 4.

En la elección de la tecnología también se ha valorado la forma de desarrollar la aplicación. Se ha considerado como requisito que se pueda acelerar el desarrollo del sistema a nivel organizativo ante una hipotética continuidad de desarrollo del sistema en el futuro, de forma

que se posibilita el trabajo en paralelo de varios ingenieros y se suaviza el impacto ante un cambio a otras tecnologías que pudieran surgir, sin que por ello se afecte a todo el diseño del *Marco* desarrollado.

La elección de la tecnología es fundamental a la hora de obtener las funcionalidades de una aplicación; sin embargo, en el desarrollo del presente proyecto este aspecto no siempre ha sido decisivo para la elección de la tecnología utilizada, ya que con algunas de las alternativas barajadas a las tecnologías escogidas se podrían también haber obtenido las mismas funcionalidades. Las ventajas principales de la tecnología escogida es la facilidad de integración de contenido multimedia en las interfaces graficas desarrolladas, así como el estado de madurez de dicha tecnología.

5.1.1 Condiciones Iniciales y Elección de la Tecnología

La elección de las herramientas software estará condicionada por cuatro grandes objetivos establecidos *a priori* como **condiciones iniciales** del sistema a desarrollar:

1. La posibilidad de modificar la configuración del servidor y, por tanto, la elección de los diferentes videos e imágenes que se mostrarán en las diferentes *ubicaciones*, a través de aplicaciones informáticas ampliamente extendidas y que vengan instaladas en cualquier ordenador personal de uso doméstico como es un navegador web.

De este modo no es necesario instalar ninguna aplicación específica desarrollada y la configuración podría realizarse desde cualquier equipo (EACCC), ya que normalmente un **navegador web** viene instalado en prácticamente la totalidad de los sistemas de usuario.

2. Realizar una interfaz lo más amigable posible hacia el administrador del sistema. Dentro de este objetivo también se busca que el administrador del sistema pueda hacerse una idea, lo más aproximadamente posible, de la configuración que realiza desde el servidor. Esta verificación debe poder realizarse sin tener que desplazarse a cada *emplazamiento físico* para ver qué y cómo se muestra el contenido que se está emitiendo.

Por tanto, se debe poder integrar imágenes y vídeos dentro de la interfaz del administrador del sistema para que se pueda establecer una **previsualización**.

3. Separación de forma modular en *diferentes capas*, ante la posibilidad de cambiar de tecnología en versiones posteriores de la aplicación, de modo que el *Marco* se mantenga con las mínimas modificaciones. Este aspecto se ha considerado importante ante una evolución del sistema. A efectos prácticos, y con el objetivo de acelerar el desarrollo del sistema, se ha supuesto la situación de que se producirá el trabajo en paralelo de varios ingenieros o desarrolladores; así, se espera que cada ingeniero pueda trabajar en una de las capas especializándose en la tecnología utilizada en esa capa, es por ello que un diseñador gráfico que programase en *ActionScript* no necesita conocer nada acerca del sistema de *bases de datos* implementado.
4. Generar tráfico de forma eficiente entre la configuración del servidor y el administrador del sistema. Esto es conveniente debido a que dentro de la interfaz de configuración se presentan imágenes y videos. Así, se espera realizar una interfaz de *configuración asíncrona* para realizar peticiones conforme a la información necesaria.

Para justificar el razonamiento de la **elección de la tecnología**, se exponen de forma resumida cada uno los puntos considerados *a priori*.

El **punto uno**, que se ha declarado como objetivo inicial, nos limita el uso de tecnologías a lenguajes interpretados por un *navegador web*. De esta forma se elimina la posibilidad de realizar la aplicación con herramientas diferentes a las de las *aplicaciones web*.

El **punto dos** descarta el uso de tecnologías web tradicionales como es el uso de *HTML* estático. Condiciona que la aplicación tenga características *RIA (Rich Internet Applications, Aplicaciones de Internet Enriquecidas)* que son un tipo de aplicaciones con más ventajas que las tradicionales aplicaciones web. Surgen como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales y nos permite introducir video e imágenes de forma dinámica.

El **punto tres** nos marca una característica muy interesante para las modificaciones futuras de la aplicación. Separar de forma modular cualquier sistema complejo convierte el problema en un subconjunto de problemas más sencillos e independientes, por una parte se puede abordar el problema general como pequeños problemas y, por otra parte, la resolución de problemas queda enmarcada en un contexto menor. De este modo el *Marco* se mantiene finalmente más robusto, frente a la eficiencia potencial de un sistema completo sin diferenciarlo en capas. La analogía más evidente es el desarrollo de protocolos aplicando un modelo de referencia, como puede ser el *modelo de referencia OSI* [73] [74], donde cada capa realiza una función específica y cambiar una u otra no afecta al conjunto de la aplicación (o sistema) que lo utiliza o las capas por debajo pueden ser utilizadas por otras aplicaciones (o sistemas).

Si se decidiera cambiar la *Interfaz Gráfica del Usuario* (la cual se ha desarrollado básicamente como veremos en *Flash*, *ActionScript 3*, *JavaScript* y *HTML*) a otras tecnologías como puede ser *Java (applet, Java SE, JavaFX,...)*, no implicará cambiar el *Motor* desarrollado en *php* y *MySQL*. La nueva implementación seguiría llamando a los archivos *php*, y estos seguirían ejecutando ciertas acciones en el servidor y devolverían contenido en formato *XML* para ser interpretado.

Con este sistema de desarrollo, de igual forma, si se quisiera cambiar la forma de acceder a la *bases de datos*, se debería sustituir el *código php* por la tecnología alternativa, por ejemplo *ASP* o *J2EE*; no obstante, no sería necesario modificar las tablas *MySQL* o apenas la interfaz del usuario si se mantuviese el mismo estándar de presentación de datos en formato *XML*.

El **punto cuatro** también limita el uso de aplicaciones web tradicionales. Normalmente en las tecnologías web *síncronas*, hay una recarga total de la página cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página ante un mínimo cambio. Así, para cumplir esta parte de las especificaciones se necesita una *tecnología web asíncrona* y la elegida, como veremos en el siguiente apartado, ha sido hacer uso de pequeñas aplicaciones *swf* programadas en *ActionScript3* combinadas con la interacción mediante otras aplicaciones (*php*, *MySQL*) de modo que se establece intercambio de datos (*xml*).

5.1.2 Elección de las Aplicaciones Software para el Desarrollo del Marco

Existen varias alternativas para poder cumplir los objetivos que se han marcado. Básicamente el objetivo es desarrollar una aplicación interpretada por un navegador web, diseñada y desarrollada en capas, y que ofrezca presentación de videos e imágenes haciendo uso de peticiones asíncronas. Existen diferentes combinaciones de tecnologías para conseguirlo, entre ellas:

- En el lado del cliente hay también diferentes opciones para ofrecer interactividad en la configuración: *Silverlight*, *Flash Player*, *OpenLaszlo*, *JavaFX*, *AJAX*,...
- En el lado del servidor existen una variedad de alternativas: *ASP (Active Server Pages)*, *JSP (Java Server Page)*, *PHP (PHP Hypertext Pre-processor)*, *Perl*, *Python*,...
- En el almacenamiento de forma persistente existen alternativas tanto en las técnicas como en las *bases de datos*, las cuales podrían ser: *Postgresql*, *mSQL*, *Driezzle*, *MariaDB*, *MySQL*, *MSSQL*, *SQLite*, *MS Access*, *Cassandra*...

Las principales diferencias que se derivarán de la elección de la tecnología escogida son el nivel de usabilidad en la *Interfaz Gráfica de Usuario* desarrollada, la velocidad de respuesta, la escalabilidad del sistema desarrollado y la dificultad del desarrollo.

Adobe Flash [75], mediante el *plugin Flash Player* para su reproducción en el navegador, históricamente ha sido la tecnología más extendida en la creación de aplicaciones *RIA (Rich Internet Applications, Aplicaciones de Internet Enriquecidas)* por lo que es considerada como una tecnología madura que puede resolver las necesidades del proyecto. El uso de *Flash Player* y el lenguaje *ActionScript 3* permite, entre otras, realizar conexiones a *socket* desde archivos compilados que se cargan y ejecutan en el navegador mediante el *plugin Flash Player*. Además, el *plugin* de *Flash Player* fue específicamente diseñado y desarrollado para la inclusión de contenido multimedia y animaciones en las páginas web, aunque actualmente también pueden ser ejecutadas como verdaderas aplicaciones en sus versiones más recientes si utilizamos *Adobe AIR* para su compilación y ejecución [76]. Además, *ActionScript 3* es un lenguaje orientado a objetos para la creación de aplicaciones sobre clientes *Flash* (tales como *Flash Player* y *Adobe AIR*). Por tanto, se consideró que la tecnología *Flash Player* a modo de *plugin* en un navegador web, junto con el desarrollo de aplicaciones *ActionScript 3* que pueden controlar la línea de tiempos de una reproducción, es ideal para desarrollar aplicaciones que muestren videos e imágenes.

Algunas características consideradas e interesantes de *ActionScript 3* son las siguientes:

- *ActionScript 3* (versión actual), mantiene una sintaxis influenciada por *Java* y *C#*. Así, el núcleo del lenguaje *ActionScript* resulta muy familiar para cualquier programador. En concreto el núcleo de *ActionScript 3* está inicialmente basado en los requisitos de lenguaje de *ECMAScript 4ª Edición*¹³, el cual estuvo en proceso de desarrollo hasta finales de 2007 [77] [78]. Así, *ECMAScript* impone la sintaxis y gramática básica de *ActionScript 3*, es decir, la sintaxis del código usado para crear cosas tales como expresiones, sentencias, variables, funciones, clases y objetos. *ECMAScript* impone, también, un pequeño conjunto de tipos de datos incorporados para trabajar con valores comunes: *String*, *Number* y *boolean* [79].
- *ActionScript 3* implementa en la medida de lo posible *ECMAScript for XML (E4X)*, estandarizado como *ECMA-357* [80] [81]. *E4X* ofrece un conjunto fluido y natural de construcciones del lenguaje para manipular *XML*. Al contrario que las *API* de análisis de *XML* tradicionales, *XML* con *E4X* se comporta como un tipo de datos nativo del lenguaje. De este modo, *E4X* optimiza el desarrollo de aplicaciones que manipulan datos *XML*, ya que reduce la cantidad de código necesario con respecto

¹³ En realidad la versión 4 de *ECMAScript* fue abandonada [164], y *ActionScript 3* está basado en los requisitos del lenguaje en *ECMA-262 3ª Edición*, pero añadiendo propuesta en los *drafts* para *ECMAScript 4ª Edición* [162]. Es por ello que puede considerarse como un lenguaje basado en *ECMAScript 4ª Edición*.

a las *API* de análisis de *XML* tradicionales [82], por tanto esta característica convierte al lenguaje *ActionScript3* en adecuado para la manipulación de datos *XML*, característica muy conveniente para la división de trabajo en niveles o capas independientes en su desarrollo.

- El código *ActionScript 3* se debe compilar en un fichero *swf*. El fichero *swf* puede incluir tanto códigos binarios como recursos integrados (gráficos, sonidos, video y fuentes). Algunos archivos *swf* contienen recursos solamente y no el código, mientras que otros contienen el código únicamente sin los recursos. Un programa individual de *ActionScript 3* puede residir completamente en un único fichero *swf* o puede estar dividido en múltiples ficheros *swf*: un archivo *swf* específico mostrará el punto de entrada al programa y cargará el resto de ficheros *swf* necesarios. Dividir un complejo programa *swf* en varios ficheros facilita su mantenimiento y, para aplicaciones de red, puede conceder al usuario un acceso más veloz a las diferentes secciones del programa. De este modo la programación de aplicaciones web mediante archivos *swf* permite la inclusión de archivos multimedia de forma eficiente.
- Debido a que *ActionScript* no viene provisto de driver para acceder a las *bases de datos*, es necesario y conveniente introducir un elemento intermedio, a modo de *gateway*, que sí tuviera esta característica implementada y al que se pudiera acceder desde las aplicaciones *swf*. El lenguaje elegido para tal fin fue *php*, el cual permite devolver código en formato *XML* que la aplicación *swf* puede interpretar. Esta necesidad casaba completamente con el criterio por el cual se divide el desarrollo de la arquitectura en diferentes capas, por lo que se adoptó finalmente como tecnología escogida para el desarrollo del *Marco para Aplicaciones de Difusión Selectiva* en la ejecución del presente proyecto fin de carrera. Así, la tecnología elegida ha sido *Flash Player*, donde se han programado y compilado diferentes archivos en *Actionscript3*, estos archivos realizan llamadas de forma *asíncrona* a *scripts* programados en *php* que realizan peticiones a la *base de datos* del servidor a través del driver para *MySQL*. Además, los *scripts* programados en *php* en muchas ocasiones van a necesitar ejecutar otros procesos, de forma que se devuelve el resultado formateando los datos en *XML* hacia la aplicación *swf* que hace una interpretación de éstos y los presenta al administrador del sistema mediante objetos *swf* embebidos en una página web.
- Se ha comentado que *ActionScript 3* permite la interacción mediante *sockets*. Aprovechando esta capacidad, se puede cargar en una página web, de forma embebida (cómo se detallará), una aplicación *swf* a modo de objeto que se puede ejecutar mediante el *plugin Flash Player*, de forma que ésta pueda estar monitorizando la información escrita sobre un *socket* y, así, hacer llamadas a funciones *JavaScript* desde las aplicaciones *swf* (clase *ExternalInterface*) que, por ejemplo, recarguen la página. Como veremos (apdo. 5.4 Sistema Software Diseñado y Desarrollado) esta capacidad es clave dentro de la tecnología y nos permitirá hacer uso de navegadores web como medio de presentación del contenido multimedia hacia el potencial espectador y nos permitirá utilizarla, por tanto, para desarrollar los *ERCs*.

Por todas estas características, se establece que las aplicaciones *swf* programadas en *Actionscript3*, junto con otras tecnologías en el lado del servidor, es la elección más adecuada para el desarrollo del *Marco*. Así, se ha optado por utilizar *Flash (ActionScript 3)* para elaborar la *Interfaz Gráfica de Usuario*, junto con *php* y *MySQL* para sustentar el *modelo de datos* y elaborar la lógica de funcionamiento de la aplicación de difusión selectiva en su conjunto.

Existen también otras tecnologías propietarias como *Silverlight (Microsoft)* [83], similar a *Flash Player (Adobe)*, la cual también han sido diseñada para la creación de RIAs. Sin embargo, *Silverlight* es una tecnología menos madura y puede considerarse de reciente desarrollo (2007), frente a *Flash Player* cuya primera versión apareció en el año 1996 [84].

Por otra parte, existen otras tecnologías que aportan mayores funcionalidades. Dentro de estas tecnologías nos encontramos, por ejemplo, con *Java*, la cual a través de la *Máquina Virtual de Java (JVM, Java Virtual Machine)* ejecuta aplicaciones completas en un navegador web. Dentro de las tecnologías *Java*, destaca *Java FX* [85], cuya tecnología ha sido desarrollada para la creación de *Rich Internet Applications (RIAs)*, como se ha comentado, aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas; no obstante, carece de aportes significativos con respecto a *Flash Player* y *ActionScript 3*.

Dentro de las alternativas que se ejecutan en un navegador, existe una técnica denominada *AJAX (Asynchronous JavaScript And XML, JavaScript Asíncrono y XML)*, la cual puede realizar la mayoría de las tareas *asíncronas* a través del objeto *xmlHttpRequest* [86] y podría cumplir con los mayoría de los requisitos, ya que podría integrarse con lenguajes del lado del servidor. Dentro de la definición más genérica y extendida de *AJAX*: “*técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications, Aplicaciones de Internet Enriquecidas)*”[87], nos da una idea del potencial esperado con el uso de estas tecnologías web. *AJAX* es por tanto una tecnología *asíncrona*, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. *JavaScript* es el lenguaje interpretado (*scripting language*) en el que normalmente se efectúan las funciones de llamada de *AJAX* mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en la mayoría de los navegadores actuales. En cualquier caso, no es necesario que el contenido *asíncrono* esté formateado en *XML*; sin embargo, ésta es una buena opción para establecer un intercambio de datos ya que está estandarizado a un cierto formato en los mensajes intercambiados y forma parte de la propia definición de *AJAX*. Se ha de indicar que las técnicas *AJAX* se pueden integrar con, básicamente, cualquier aplicación en el lado del servidor [88] [89], pero complica el manejo de contenido multimedia con respecto a *Flash Player* y *ActionScript 3*.

5.1.3 Servidor Web en los Equipos Principales: Apache, MySQL y PHP

Se ha comentado la necesidad de establecer algún mecanismo por el cual, de forma remota, sea posible configurar tanto el *ESCC* como los *ERCs*. Como veremos, la configuración del *ESCC* condicionará el funcionamiento de todo el sistema, mientras que la configuración de los *ERC* será, básicamente, para que estos equipos obtengan conectividad al *ESCC*.

Un requisito de partida para el *Marco* es eliminar la necesidad de instalación de aplicaciones especiales en los *EACCC* (solo un navegador web), de modo que prácticamente cualquier equipo se pueda usar como *EACCC* para realizar las configuraciones del resto del sistema. Sí será necesario, sin embargo, instalar ciertas aplicaciones en los *ERCs* y *ESCC*, para proporcionar a los administradores del *Marco* la funcionalidad de configuración de estos equipos de forma remota a través de un navegador instalado en el *EACCC*.

Para ello, se ha determinado que es necesario que los equipos se comporten como auténticos servidores web y ejecuten determinadas acciones en base a cierta configuración almacenada. Como decisión de diseño, se ha decidido que tengan instaladas las siguientes aplicaciones [90]:

- **Apache:** Es la aplicación que convierte a los equipos en servidores web. Su principal función es la de procesar cualquier archivo que se haya solicitado desde un navegador y mostrar el resultado en función del código del archivo.
- **PHP (PHP Hypertext Pre-processor):** Es el lenguaje que nos permite crear aplicaciones online dinámicas realizando consultas al servidor. Su sencillez y precisión a la hora de trabajar lo convierten en uno de los lenguajes más utilizados para este fin y muy conveniente para el propósito del sistema desarrollado.
- **MySQL:** Se trata de un servidor SQL (*Structure Query Language, Lenguaje de Consulta Estructurado*) que puede procesar consultas complejas. Una característica útil, es su capacidad de trabajar como sistema de *base de datos* relacional, esto significa que se pueden combinar tablas diferentes para optimizar la velocidad de procesos y la eficacia a la hora de trabajar.

La combinación de estas tecnologías es usada principalmente para definir la infraestructura de un servidor web utilizando un paradigma de programación *cliente-servidor* para el desarrollo del *Marco de Aplicaciones*. A pesar de que el origen de estos programas de *código abierto* no ha sido específicamente diseñado para trabajar entre sí, la combinación se ha popularizado debido a su bajo coste de adquisición. Por tanto, cuando se combinan representan un conjunto de soluciones que soportan servidores de aplicaciones y son asequibles desde un punto de vista de costes.

Actualmente existen diversos proyectos que instalan estas aplicaciones de forma conjunta, de modo que se suelen denominar servidores AMP (*Apache, MySQL y PHP*¹⁴). Dependiendo del *sistema operativo* en el que se instalan estas aplicaciones, se antepone normalmente la inicial L (*Linux*) o W (*Windows*): LAMP o WAMP. Sin embargo, existe un proyecto que es independiente de la plataforma donde se instale, éste proyecto se denomina XAMPP [91]. XAMPP es un servidor software libre que consiste principalmente en la *base de datos MySQL*, el servidor web *Apache* y los intérpretes para lenguajes de *script: PHP y Perl*. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), *Apache, MySQL, PHP y Perl* como aplicaciones instaladas. Incluso grandes empresas como IBM llegan a realizar comunicados [92] de las virtudes de de la aplicación XAMPP. El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para los sistemas operativos *Microsoft Windows, GNU/Linux, Solaris, y MacOS X*.

Desde el punto de vista del *proyecto fin de carrera*, una característica muy importante ha sido el hecho de desarrollar la arquitectura lo más independientemente posible del *sistema operativo*¹⁵ en el cual resida cada uno de los elementos del sistema (*ESCC y ERC*). Esto es debido a que inicialmente no estaba decidido sobre qué *sistema operativo* descansaría el sistema ya que esta característica no estaba marcada a priori (ver apdo. 5.1.1 Condiciones Iniciales y Elección de la Tecnología).

Por tanto, como se debía realizar el desarrollo de la interfaz de usuario bajo *Windows* o *MacOS X* (debido a que son los únicos sistema donde se podían compilar los archivos *swf*) y el uso de XAMPP solamente requiere descargar y ejecutar un archivo *zip, tar, o exe*, se consideró

¹⁴ El lenguaje que ejecuta los scripts no se ha considerado que sea el mismo para todos los servidores de tipo AMP y es posible identificar servidores clasificados de este modo, pero utilizando otros lenguajes como *Pyhton* o *Perl*.

¹⁵ Otras tecnologías consideradas obtienen una compatibilidad absoluta, de modo que no habría que realizar ningún tipo de cambio al pasar de un sistema operativo a otro (considérese *Java*), no obstante, una vez elegida la tecnología hemos de indicar que este factor es importante al iniciar el desarrollo del *Marco* diseñado.

la alternativa apropiada para iniciar el desarrollo. Además, *XAMPP* es un proyecto muy activo y se actualiza regularmente para incorporar las últimas versiones de *Apache*, *MySQL*, *PHP* y *Perl*. También incluye otros módulos como *OpenSSL*, el cuál ha sido útil para proporcionar conexión segura a través del navegador, y *phpMyAdmin*, que ha sido utilizada en el desarrollo como herramienta de apoyo, sobre todo para '*Troubleshooting*' (detección y resolución de errores).

Para instalar *XAMPP* se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado; no obstante, finalmente el resultado es el mismo [20] que instalarlas de forma independiente. El desarrollo de una aplicación sobre un servidor *Windows/Linux/MacOS X* en el que se haya instalado *XAMPP*, se basará en la programación de los diferentes *script* (en el caso concreto del presente proyecto se ha programado en el lenguaje *php*, aunque existían otras alternativas como *Perl*) y en la configuración de los archivos que describen el comportamiento de estas aplicaciones. Así, realizar la migración de un *sistema operativo* a otro de una aplicación construida bajo *XAMPP*, no es más que copiar los archivos programados al nuevo servidor y realizar las mismas configuraciones en el servidor *Apache*, la *base de datos MySQL* y la aplicación que ejecuta los *scripts*, en nuestro caso *php*. Adicionalmente, con el paquete ejecutándose bajo *Linux* se incluye una herramienta especial para proteger fácilmente las partes más importantes de las aplicaciones a nivel de seguridad (*lampp security*), el cual es básicamente un *script* que realiza algunas configuraciones sobre las aplicaciones.

5.2 Resumen del Marco para Aplicaciones de Difusión Selectiva Implementado

En función de las características del sistema final y del comportamiento de la *Arquitectura Funcional* de la *Aplicación* se ha diseñado un *Marco* en el que se separa el sistema en módulos que realizan determinadas acciones.

En concreto, en el desarrollo para abarcar el *Marco* se han establecido dos diseños. Por una parte, el del *Marco* establecido para *configurar* los sistemas a través de la *arquitectura funcional* planificada y, por otra, el diseño de los módulos que proporciona la *sincronización* y *reproducción* del contenido multimedia.

5.2.1 Marco Diseñado y Desarrollado para Establecer la Configuración

En la Ilustración 32 se muestra un esquema general del *Marco* para proporcionar la configuración tanto al *ESCC*, como al *ERC*.

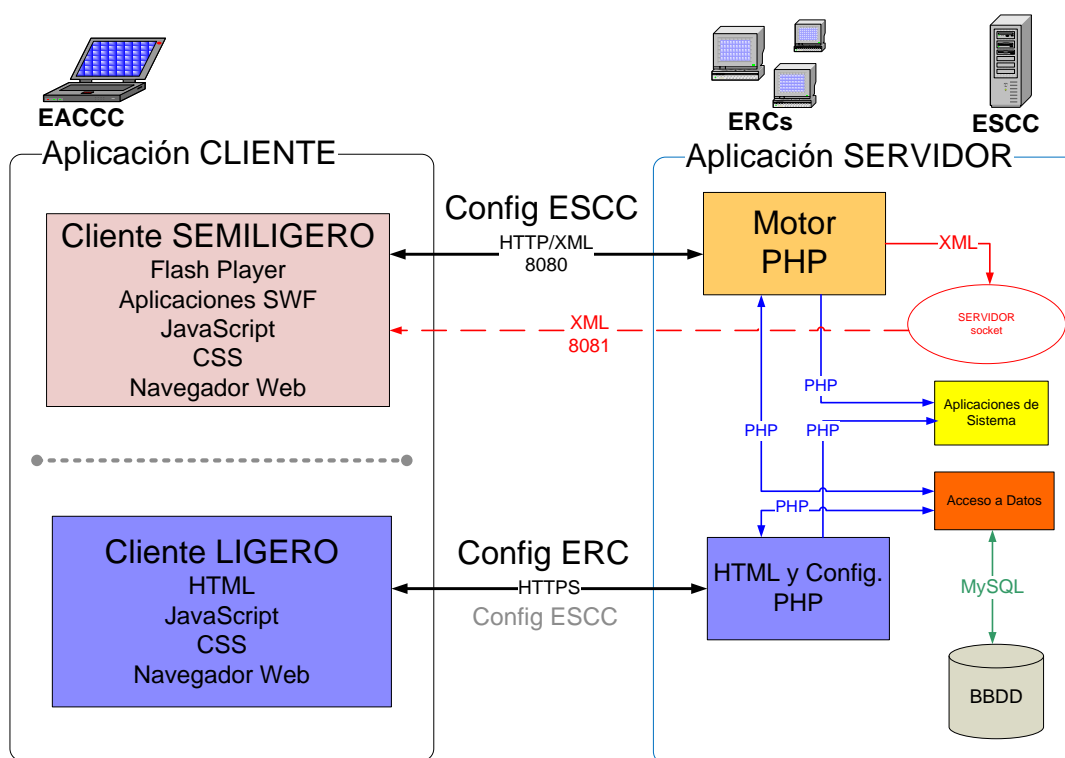


Ilustración 32.- Marco diseñado y desarrollado para establecer la Configuración de los equipos.

Se han representado cada una de las aplicaciones desarrolladas tanto en el *ESCC* como en el *ERC*. En la parte superior del esquema se representan los equipos donde se ejecutan las aplicaciones.

En la parte *superior izquierda* (**Aplicación CLIENTE**) se indica que la configuración se realiza desde los equipos *EACCC* y que las aplicaciones embebidas y el código interpretado se ejecutan, por tanto, sobre los clientes *http* (*navegadores web*) desde los que se realiza la configuración (*Interfaz Gráfica de Usuario web*); no obstante, aunque se ejecuten en los *EACCC*, estos archivos y aplicaciones están almacenados de forma permanente en los *ERC* y el *ESCC*, de modo que se implementa una *arquitectura Cliente-Servidor*.

- **Cliente SEMILIGERO:** Se ha denominado así a la ejecución del cliente que configura el ESCC debido a la inserción de *aplicaciones swf* en el código *html* recibido. Estas *aplicaciones swf* se cargan de forma embebida (ejecutadas por el plugin *Flash Player*) e introducen inteligencia a la aplicación en el lado del navegador. Además, gestionan el procesamiento de la información que se presenta al usuario de la aplicación web. No obstante, debido al poco procesamiento introducido y a que no hay que realizar ninguna instalación de estas aplicaciones, se considera como un *cliente semiligero*. Para poder hacer un uso correcto del *Cliente SEMILIGERO* se deberá tener instalado el *plugin Flash Player* en el navegador desde el que se quiera establecer la configuración. Estas aplicaciones se cargan a través del navegador en el EACCC cuando se accede a la configuración del ESCC (código *html*) y es el *plugin* el encargado de interpretar la lógica de las aplicaciones *swf*.

La estrategia más significativa y diferenciadora con respecto al *Cliente LIGERO* (configurador íntegro del ERC) es la de utilizar *peticiones asíncronas* para recargar los datos exclusivamente necesarios en la *Interfaz Gráfica de Configuración del Usuario*. Otra característica importante es el acceso directamente al *servidor socket*, de modo que la *Interfaz Gráfica de Usuario* es capaz de evaluar las órdenes que allí se han establecido y, así, se puede realizar una pequeña monitorización del sistema.

- **Cliente LIGERO:** El acceso a la configuración de los ERCs, a través del navegador, carece de *aplicaciones swf* embebidas que proveen inteligencia a la configuración. De esta forma la ejecución se realiza en el servidor y en el cliente tan sólo se realiza la interpretación de los datos para mostrarlos sobre el navegador. El *Cliente LIGERO* se carga cuando se accede a la configuración¹⁶ del ERC, de modo que cada vez que se da una orden de configuración se hace una llamada al propio *script php* que devuelve la información con los datos actualizados de las *tablas* de la *base de datos*.

En la parte superior derecha (**Aplicación SERVIDOR**) se representan los equipos donde se ejecutan los *script* y aplicaciones. Estas aplicaciones y configuraciones se ejecutan directamente en los equipos que se está administrando. Se observa en el ERC y el ESCC que existen comunicación interna entre las diferentes capas definidas. Esto establece independencia entre capas, las cuales nos permiten un desarrollo independiente. Se observa, también, que toda la inteligencia para proporcionar la configuración del sistema se encuentra ejecutándose en los servidores (ERC y ESCC). Para la comprensión del esquema de la Ilustración 32 se ha de tener en cuenta también que, aunque se han representado algunas capas con un solo símbolo en los ERC y ESCC, corresponden a elementos independientes; es decir, la *base de datos (BBDD)* y, por tanto, el *Acceso a Datos* son totalmente independiente en el ERC y ESCC, ya que habrá una estructura de *tablas* de la *base de datos* diferente en el ERC a las que se han almacenado en el ESCC de acuerdo a las necesidades de configuración de cada equipo. Lo mismo sucede con la capa que se ha denominado *Aplicaciones de Sistema* que, aunque pueden llegar a ser muy similares, son diferentes y personalizadas a las necesidades de cada tipo de equipo.

El *Motor PHP* es el encargado de establecer y proveer la configuración al *Cliente SEMILIGERO*. Recibe *peticiones http* con parámetros pasados mediante los métodos *GET/POST* y devuelve las diferentes configuraciones en formato de archivos *XML*. Estos archivos devueltos son interpretados por la *aplicación swf* y mostrados en su mayoría hacia el usuario que está configurando el *Marco*. El *Motor PHP* es el encargado de escribir en el *Servidor socket* en función de la lógica establecida ante las *peticiones* recibidas.

¹⁶ El método *Cliente LIGERO* también es utilizado en configuraciones de parámetros del sistema del ESCC donde es prescindible el uso de *aplicaciones swf*.

El bloque *HTML y Configuración PHP* establece las configuraciones en la *base de datos (BBDD)* para el *ERC*. Las funciones son similares a las realizadas por el *Motor PHP* para la recepción de peticiones, pero devuelven el código formateado como *html* directamente, generando la página web de configuración al estilo tradicional¹⁷. El bloque *HTML y Configuración PHP* no tiene interacción con el *Servidor Socket*.

Se observa también dos módulos definidos en la Ilustración 32, los cuales tienen comunicación unidireccional: *Servidor Socket* y *Aplicaciones de Sistema*. Estos módulos sirven para realizar tareas que afectan al sistema de forma general: son utilizadas para realizar cambios en la configuración del equipo (*Aplicaciones de Sistema*) o comunicaciones con el resto de equipos (*Servidor Socket*). En esta comunicación unidireccional no se espera respuesta y son consideradas como elementos para establecer órdenes e instrucciones a los diferentes equipos de la arquitectura.

5.2.2 Marco Diseño y Desarrollo para Proporcionar el Funcionamiento (Sincronización y Reproducción)

Dentro del *Marco* existe una interacción entre los diferentes módulos no asociada a establecer una configuración en el sistema, sino para mostrar el contenido multimedia desde los *ERCs* a través de los *SPC (Sistema de Presentación de Contenido)* según se haya configurado en el *ESCC*.

Es, por tanto, que, una vez configurado el sistema, aplica un *Marco* subyacente que permite hacer funcionar el sistema en base a la configuración establecida. El esquema de este *Marco* se presenta en la Ilustración 33.

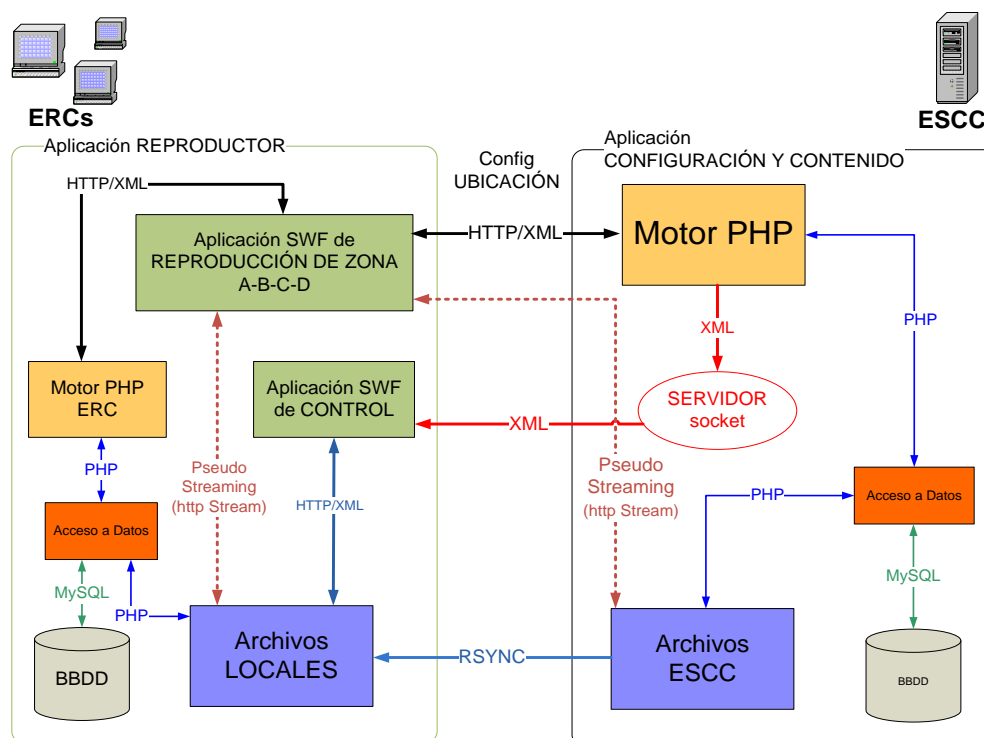


Ilustración 33.-. Marco diseñado y desarrollado para el Funcionamiento de la Aplicación de Difusión Selectiva.

¹⁷ Al ser páginas *html* sencillas, no se ha tenido la necesidad de aplicar ni técnicas AJAX, ni un patrón de arquitectura *software* del tipo *Modelo Vista Controlador* [157].

Describimos ahora las aplicaciones que se ejecutan en cada uno de los equipos. Por una parte, se tiene la *Aplicación REPRODUCTOR* que se ejecuta en los *ERCs* y, por otra, se mantiene la *Aplicación CONFIGURACION Y CONTENIDO*, que se ejecuta en el *ESCC*, y que proporciona tanto la configuración establecida por el administrador del sistema como los videos e imágenes que se han subido al servidor.

Dentro de cada una de las aplicaciones principales que forman el *Marco* se tiene, en la parte izquierda un conjunto de elementos o módulos que representan la **Aplicación REPRODUCTOR**. A continuación se presenta una breve descripción de los elementos que la componen, los cuales se ejecutan en el *ERC*:

- **Motor PHP, Acceso a Datos, BBDD (MySQL):** Mediante la combinación de estos elementos se obtiene los parámetros necesarios para obtener la conectividad con el *ESCC*. Esta conectividad tiene como objetivo inicial recuperar la configuración del *Escenario* para la *Ubicación* que ha sido programado en el equipo. Estos elementos también permiten reconocer si el contenido multimedia está almacenado de forma local para, así, aplicar el mecanismo definido para la arquitectura *Auto-Proxy*.
- **Aplicación SWF de Reproducción de Zona:** Esta aplicación esta embebida sobre la página web que presentará el *Escenario* y reproducirá los archivos en función de la configuración de las dimensiones y posición de las *Zonas* sobre la pantalla emisora. Esta aplicación mantiene la lógica de reproducción de los archivos multimedia. Al incluirla de forma embebida (en la página web mostrada a pantalla completa) se le pasarán una serie de parámetros entre ellos la *Ubicación*, la *Zona* a la que corresponde y el servidor al cual debe realizar las peticiones. Con esta información, la aplicación realiza una serie de acciones:
 - Realiza una petición *http* al *ESCC* y recibe la *lista de reproducción* actual según la fecha en formato *XML*. En la respuesta, además del nombre e intervalo de reproducción de cada archivo de la lista, se obtiene la fecha para la cual se debe establecer la nueva petición de la siguiente lista de reproducción.
 - La *Aplicación SWF de Reproducción de Zona* obtiene, por tanto, el listado con los archivos. De este modo realiza una consulta *http* al servidor local (*localhost*) y verifica (*Motor PHP* en el *ERC*) cuáles de ellos se tienen almacenados de forma local y cuáles no.
 - La aplicación (*Motor PHP*) etiqueta los archivos en función de si los tiene almacenados de forma local o no, para realizar la petición directamente al *ESCC* o hacerlo de forma local cuando se establece el *Pseudo Streaming (HTTP Streaming)* obteniendo la arquitectura funcional *Auto-Proxy*.
- **Aplicación SWF de CONTROL:** Esta aplicación controla fundamentalmente el estado de *servidor socket*. Con ella es posible recoger las órdenes emitidas por el *ESCC* y actuar según la programación de cada orden. La *Aplicación SWF de CONTROL* podrá realizar dos clases de acciones: llamadas a *funciones JavaScript* que afectan a la visualización del navegador y llamadas a *script* que gestionan la actualización de los archivos locales para su sincronización.
- **Archivos LOCALES:** Es la combinación de un *script* y la programación de éste a intervalos regulares para que verifique que los archivos multimedia no han cambiado en el servidor. Si detectara que se han incluido o eliminado algún archivo o si ha sufrido una modificación, los copiará de nuevo de forma local en el *ERC*. La sincronización se realizará, bien cuando se detecten cambios, o bien cada *30 minutos* (parámetro establecido). Así, mediante esta sincronización de los archivos, la *Aplicación SWF de Reproducción de Zona* podrá realizar *HTTP Streaming* directamente sobre los archivos que estén marcados como disponibles en local (*Archivos LOCALES*).

En la parte derecha de la Ilustración 33, se representan los elementos que forman la **Aplicación CONFIGURACION Y CONTENIDIO**. Estos elementos se ejecutan en el ESCC, y proveen tanto el contenido multimedia, como la configuración a los ERCs. A continuación se presenta un resumen de los elementos más significativos:

- **Motor PHP:** En el contexto del ESCC para proveer la *configuración* al ERC, la funcionalidad es la de proveer la configuración a cada *Zona*¹⁸. Atiende peticiones *http* (TCP 8080) y devuelve código *XML*. Según se realicen configuraciones de forma paralela mediante la interfaz de configuración, el *Motor PHP* puede escribir órdenes en el *socket* que deben ser interpretadas por el ERC.
- **Acceso a Datos/BBDD:** Tiene una fuerte interacción con el *Motor PHP*, ya que según sean las peticiones se recuperarán ciertos datos de las *tablas* de la *base de datos*.
- **Servidor Socket:** Es el encargado de transmitir las diversas órdenes a los ERCs. Cuando se escribe en el *servidor socket*, se propaga la orden a cada uno de los equipos conectados al *socket*.
- **Archivos ESCC:** Se ha establecido un servicio de *sincronización* de contenido multimedia en el ESCC, con el resto de equipos ERC, para implementar la *arquitectura Auto-Proxy* definida. Cuando se carga un archivo en el ESCC, se almacena en un directorio que permite al resto de equipos descargarlo a través de un *cliente rsync*. Este directorio también estará disponible para establecer el modo *HTTP Streaming* directamente sobre el archivo desde los ERCs, cuando no lo tengan almacenado localmente.

¹⁸ Recordemos que la *Zona* es la sección rectangular que reproduce el contenido multimedia programado en el ESCC. Como detallaremos, en la *Zona* se ejecuta una aplicación que se comunica con el *Motor PHP* para obtener las *listas de reproducción* de contenido multimedia.

5.3 Entorno de Desarrollo

En este apartado vamos a describir el entorno de desarrollo que hemos utilizado en el proyecto. La elección de la tecnología de *aplicaciones swf* para la *Interfaz Gráfica del Usuario administrador* imponía el uso de *Microsoft Windows* para su programación (otra alternativa hubiera sido *Mac OS*). Esto es debido al uso de *Flash* y *ActionScript 3* (*Adobe Flash CS3 Professional*) para obtener las aplicaciones *swf* que permitirán trabajar con videos e imágenes de forma eficaz [93].

Sin embargo, y por motivos económicos, el escenario objetivo de despliegue es que tanto el *ESCC* como el *ERC* se ejecuten en sistemas *Linux* [94]. Para probar sobre un entorno representativo del despliegue real esperado, pero sin complicar el entorno de desarrollo requiriendo varios equipos físicos distintos, se decidió utilizar *máquinas virtuales* [95] [96] [97] para crear equipos virtualizados *ESCC* y *ERC* donde se almacenan los archivos de la *Interfaz Gráfica de Usuario* y sobre los que se trabajaría en red. Con este esquema es posible desarrollar sobre *Windows* y, al mismo tiempo, probar el despliegue de los sistemas sobre *Linux* utilizando un único equipo físico que representa varios equipos en red.

El esquema de este escenario para el desarrollo se muestra en la Ilustración 34 y, como veremos en los diferentes capítulos, ha sido el entorno de trabajo clave en el desarrollo del proyecto también a la hora de realizar las pruebas:

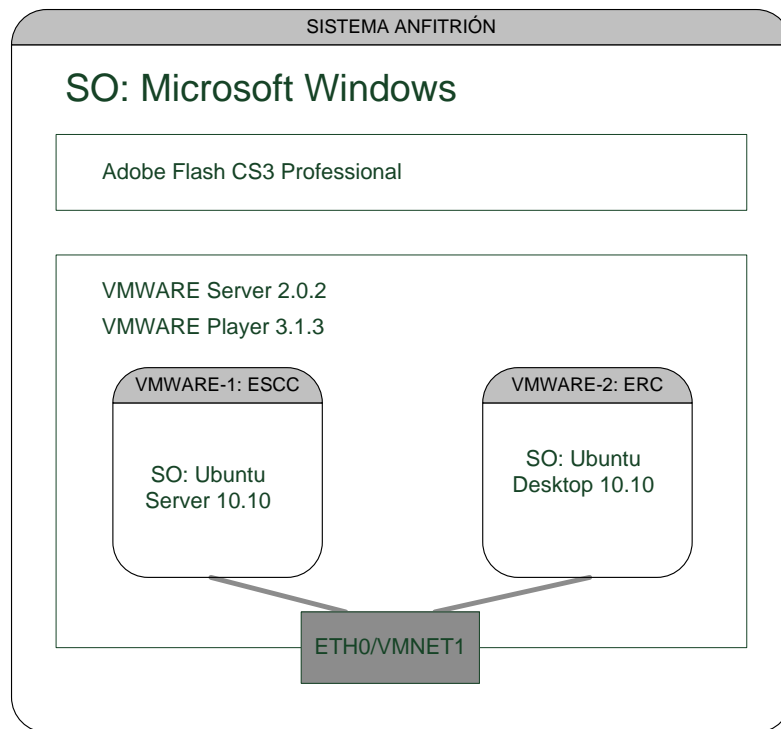


Ilustración 34.- Equipo anfitrión con varias instancias virtualizadas de los equipos ESCC y ERC.

Este esquema de trabajo es realmente útil debido a que el modo de operar es bastante realista en comparación a una situación en la que se tienen diferentes equipos donde se programan diferentes prototipos. Es decir, a nivel lógico de conectividad es como si se estuvieran desarrollando los equipos principales (*ESCC*, instancia de *VMWARE-1* y *ERC*, instancia de *VMWARE-2*) desde un tercer equipo, desde el que nos conectamos (*SISTEMA ANFITRIÓN*) mediante diferentes métodos de conectividad y aplicaciones. La conectividad es

obtenida a través de una interfaz virtual (VMNET1) o a través del interfaz físico (ETHERNET/WIFI).

Por tanto, para establecer un escenario cómodo de trabajo, sobre las máquinas virtuales (*ESCC* y *ERC*) [95][55], se han instalado algunas aplicaciones (*VMware Server*, *VMWare Player*) que permiten el acceso a los archivos y a las configuraciones de forma directa desde la máquina anfitriona. El sistema anfitrión es gobernado por un *sistema operativo Microsoft Windows XP* y, los equipos a desarrollar, *ESCC* y *ERC*, se estableció que ejecutaran el *sistema operativo Ubuntu Server 10.10* y *Ubuntu Desktop 10.10* respectivamente.

Las aplicaciones relevantes instaladas sobre las máquinas virtuales han sido:

- **Samba:** Nos sirve para acceder directamente a los archivos a través de la red. Se configuran las máquinas virtuales para compartir los archivos del proyecto y, así, puedan ser editados (los archivos *Flash* y, por tanto, la programación *ActionScript3*) directamente desde la máquina anfitriona basada en *Windows*.

```
sudo apt-get install samba
```

La configuración realizada (`/etc/samba/smb.conf`) sobre ambos equipos se muestra a continuación:

```
[public]
comment = div-dat publico (no recomendado)
path = /opt/lampp/htdocs/div-dat
public = yes
writable = yes
```

Se ha de tener en cuenta que esta configuración es para establecer un sistema de desarrollo del *Marco* finalmente desarrollado y que, desde el punto de vista de seguridad, no es del todo apropiado ya que no se ha requerido, por comodidad ningún control de acceso. Sin embargo, es muy conveniente para abrir archivos en red y poder realizar el desarrollo desde diferentes sistemas operativos a través de la aplicación *samba*. También se ha de indicar que se han de establecer los permisos adecuados a los archivos de las carpetas a compartir.

- **Openssh:** para realizar configuraciones en los equipos se permite el acceso a través de clientes *ssh* sobre los equipos. La instalación de la aplicación se realiza sobre ambos sistemas virtualizados y se mantiene la configuración por defecto que aplica al instalar *openssh-server*:

```
sudo apt-get install openssh-server
```

Se puede modificar esta configuración en el archivo `/etc/ssh/ssh_config`; no obstante, la configuración por defecto es suficiente para poder realizar pequeñas modificaciones sobre los distintos archivos sin tener que abrir la consola de *VMWare*.

Las aplicaciones que se han presentado no son necesarias para el funcionamiento del *Marco*, aunque sí se han realizado para establecer un método de trabajo en la configuración y desarrollo del *Marco*. Existen otras alternativas que establecen parte del proceso de forma automática, como el *Control de Versiones* el cual requiere instalar algún programa adicional (*CVS*, *Subversion*, *SourceSafe*, *ClearCase*, *Darcs*, *Bazaar*, *Plastic SCM*, *Git*, *Mercurial*,...) [98]; sin

embargo, debido a que sólo se establecían las modificaciones desde la *máquina* o *sistema anfitrión* y un solo programador, no se consideró necesario ejercer esta alternativa.

5.4 Sistema Software Diseñado y Desarrollado

Se ha de tener en cuenta que dentro del planteamiento de la implementación del *Marco* se han desarrollado aplicaciones para obtener las funcionalidades necesarias en un sistema de *Difusión Selectiva*. Adicionalmente, dentro del *software* desarrollado se incluirán algunas anotaciones sobre la integración de estas aplicaciones con otras existentes. Las aplicaciones de las que se ha hecho uso, aunque propiamente no se han implementado, se considera que son parte importante del sistema y es necesario mostrar las configuraciones realizadas para, así, entender el sistema completo. Así, las aplicaciones desarrolladas serán ‘piezas’ diseñadas a medida para su integración con otras aplicaciones que proveen otras funcionalidades diferentes y que, en conjunto, proporcionan un *Marco para Aplicaciones de Difusión Selectiva*.

5.4.1 Introducción a las Aplicaciones Presentes en los Equipos Principales (ESCC, ERC) y su Comunicación

Antes de describir los detalles técnicos del sistema final, se va a mostrar un resumen del método de comunicación de los diferentes equipos y también un resumen de las necesidades de software en los equipos principales. De este modo es más intuitivo reconocer, en cada una de las aplicaciones, las funciones realizadas dentro de la *arquitectura funcional* y el *Marco* planteado.

5.4.1.1 Método de Comunicación entre Equipos Principales (ESCC, ERC)

En un sistema de *Difusión Selectiva* es prácticamente imprescindible idear un sistema de comunicación entre los elementos que forman el sistema. Es posible que los equipos que presenten el contenido multimedia (ERCs), mediante un *Sistema de Presentación de Contenido* (SPC), deban comunicarse con servidores (ESCC) o, dependiendo del sistema, sensores que proporcionen información. Por esta razón, el sistema de comunicación será un elemento importante dentro de la arquitectura ya que éste determina algunas decisiones de diseño.

En el *Marco* se ha decidido que, la comunicación de las órdenes, se establezca a través de *sockets*; ya que es una interfaz de comunicación común y que se podría adaptar, en principio, a otras tecnologías. Las órdenes emitidas instan a los equipos que las reciben a realizar determinadas tareas y ejecutar determinados procesos. Así, el ESCC ‘escribe’ las órdenes en un *socket* donde los ERCs estarán ‘escuchando’ para recibir información. De esta manera, los ERCs pueden interpretar esas órdenes recibidas. El *socket* se centraliza y gestiona en el ESCC de modo que lo denominamos **servidor socket** y éste será el elemento encargado de distribuir las órdenes a los distintos clientes.

Las aplicaciones que interpretan las órdenes escritas en el *socket* se han programado en *ActionScript 3* y se ejecutan como aplicaciones *swf*. En el ERC, la forma de mostrar el contenido multimedia es a través de una aplicación web en un navegador *Firefox* a pantalla completa y las aplicaciones que controlan la información escrita en los *socket* son aplicaciones *swf* que se ejecutan en cada ERCs; en concreto, se cargan de forma embebida como objetos en la aplicación web que presenta el contenido multimedia (ver apdo. 5.2 Resumen del Marco para Aplicaciones de Difusión Selectiva Implementado).

5.4.1.2 Necesidades Software en los Equipos Principales (ESCC, ERC)

En este subapartado se presentan las necesidades en cuanto a aplicaciones y pasos de configuración de los equipos en la *arquitectura funcional* del sistema de *Difusión Selectiva*, en

concreto en el *ESCC* y en el *ERC*¹⁹. Esto nos permitirá tener una visión general de las aplicaciones que se deben ejecutar. Es decir, vamos a definir las configuraciones y aplicaciones necesarias para proporcionar el funcionamiento descrito en el apartado 4.3.1 Funcionamiento General del Sistema Sobre la Arquitectura Funcional.

Equipo Servidor de Configuración y Contenido

Al iniciar el servidor (*ESCC*) es necesario que se ejecuten la siguiente secuencia de arranque de servicios, con las configuraciones correspondientes:

- *Servidor Web:*
 - *Apache*
 - *Php*
 - *MySQL*
- *Servidor NTP*
- Se carga en las interfaces de red, la *Configuración de Red* almacenada en *tablas de la base de datos*
- Se carga la configuración *NTP* almacenada en las *bases de datos* y se auto-configura el equipo como servidor *NTP* hacia los *ERCs*
- Se ejecuta el *servidor de archivos de política de socket*
- Se ejecuta el *servidor socket* de órdenes a los *ERCs*
- Se ejecuta el *servidor para sincronización de contenido* de forma remota para proveer la funcionalidad *Auto-Proxy*.

Una vez que estos servicios se están ejecutando, el *ESCC* está dispuesto para recibir peticiones de contenido desde los *ERCs* y proveer la configuración adecuada a cada equipo. Además, está disponible para ser accedido vía un navegador web y, así, poder cambiar la configuración a través de una aplicación web.

El *servidor de archivos de política de socket* consiste en un servidor que atiende peticiones en un determinado puerto (*TCP 843*) y provee, a las aplicaciones *swf*, la configuración con las políticas de acceso que tienen estas aplicaciones a los diferentes *sockets* del *ESCC* (ver apartado 5.4.3.4.2.1, pág. 152). La política puede ser restrictiva, por ejemplo, restringiendo el acceso a algunas *direcciones IPv4* o a un *dominio* determinado. En el ámbito del *Marco* se ha creado un ***archivo de políticas de socket*** totalmente permisivo para las aplicaciones *swf*.

El *servidor socket* se encarga de controlar las conexiones y enviar los datos (órdenes) a cada *ERC*.

El *servidor para la sincronización de contenido de forma remota* permite a los *ERCs* descargarse el contenido multimedia de forma local, para no tener que realizar peticiones al servidor cada vez que se necesite el archivo para su reproducción.

Equipos Reproductores Cliente

Al igual que el *ESCC*, el *ERC* se debe configurar como un servidor web que ofrezca una *Interfaz Gráfica* para restablecer su propia configuración. Además, el *ERC* debe proporcionar otras funcionalidades como es la salida gráfica de videos e imágenes y utilizar aplicaciones

¹⁹ En los equipos *EACCC*, también se ejecutarán aplicaciones *swf* que se conectan al *servidor socket*; sin embargo, en este caso no se interpretarán dichas órdenes y serán simplemente elementos informativos para el usuario administrador del sistema. Es decir, el administrador del sistema tendrá visibilidad de las órdenes enviadas desde el *ESCC* a los *ERCs* a través del *servidor socket*.

cliente (*ntpdate*, *rsync*) para comunicarse con el *ESCC*, para sincronizar la hora o el contenido multimedia de forma local. Con ello el *ERC* debe proporcionar la siguiente secuencia de arranque de aplicaciones y configuraciones correspondientes:

- *Servidor Web*
 - *Apache*
 - *Php*
 - *MySQL*
- Se carga en las interfaces de red, la *Configuración de Red* almacenada en las tablas de la *base de datos*
- Se carga la información almacenada sobre el *ESCC* (*dirección IPv4* y *ancho de banda* de conexión para la sincronización de contenido) almacenada en las *tablas* de las *base de datos* local.
- Se ejecuta el *cliente NTP* que se conecta al *servidor NTP* (obtenido de la *base de datos* como paso previo) del *ESCC* para sincronizar la hora.
- Se ejecuta el *cliente para sincronización de contenido multimedia* (vídeos e imágenes) con el *ESCC*.
- Se ejecuta un *cliente navegador web a pantalla completa* que proporciona el medio para presentación de contenidos multimedia según la configuración establecida.
- Se ejecuta el *cliente socket* que se conecta al *servidor socket* de órdenes ejecutado en el *ESCC*.

A través de la *Interfaz Gráfica de Usuario*, que está diseñada para ser amigable, se pueden realizar diversas configuraciones sobre el *ERC*. Entre las configuraciones que se establecen a través de esta interfaz se encuentran:

- *dirección IPv4, máscara y router por defecto* de la interfaz de red (*configuración de red*).
- *dirección IPv4* del *ESCC* al que el *ERC* debe conectarse para obtener:
 - la configuración específica del *ERC*
 - el contenido multimedia del sistema
 - conexión al *servidor de socket* para recibir las órdenes
- *ancho de banda* para la sincronización del contenido multimedia con el *ESCC*

Así, al iniciar, el *ERC* adopta la *configuración de red* almacenada en la *base de datos*, obtiene la *dirección IPv4* del *ESCC* al cual se tiene que conectar e inicia una *sincronización del contenido multimedia* marcada por el *ancho de banda* también almacenado en la *base de datos*.

El equipo sincroniza la hora mediante el protocolo *NTP*, teniendo en consideración al *ESCC* como equipo servidor de hora.

Al iniciar el equipo, éste se conecta a través de un navegador web a pantalla completa a una aplicación web local que está continuamente monitorizando que el *ESCC* esté activo. Una vez que determina que se puede conectar a él, envía una petición de conexión a una aplicación escrita en *php* que devuelve la configuración en función de la *dirección IPv4 origen* del *ERC*.

El *ERC* obtiene la *configuración de la programación* desde el *ESCC* con información de la *Ubicación* a la que pertenece el equipo y el contenido multimedia que se debe difundir.

En la página web generada por el *ESCC* y cargada en el navegador del *ERC* a pantalla completa para mostrar el contenido multimedia, existe una aplicación *swf* que monitoriza continuamente el estado del *ESCC* a través de la conexión al *servidor socket*. El *ERC* se conecta al *servidor socket* en el *ESCC* de modo que interpreta las órdenes que allí se escriben. Las órdenes se escriben en función de la configuración del *ESCC* relativa a la presentación del contenido multimedia del *ERC* y, dependiendo de las órdenes, se realizan las acciones correspondientes.

Por tanto, se puede apreciar, en estas dos secuencias resúmenes del funcionamiento de los equipos, que existen algunos elementos comunes en ambas clases de equipos principales: *ESCC* y *ERC*. En los sucesivos apartados se detallan las configuraciones y aplicaciones desarrolladas para establecer las dos secuencias descritas en este apartado tanto para el *ESCC*, como para el *ERC*.

5.4.2 Configuración y Aplicaciones Comunes en los Equipos Principales (ESCC, ERC)

La configuración de varios parámetros de sistema se realizará a través de una *interfaz grafica de usuario web* por lo que se ha de transformar los equipos en pequeños servidores web, como se indicado en varias ocasiones. Además de este punto de partida común (en la configuración de los equipos) se tienen otros aspectos que se comparten en ambos equipos (ESCC, ERC):

- La *versión del software Apache, MySQL y PHP* puede ser la misma en los dos tipos de equipos.
- La *estructura general* de las páginas de presentación para la configuración web, escritas en lenguaje *php* (*archivos php base*), es similar en todos los elementos de configuración, ya se encuentren en el ESCC o en los ERCs. Dentro de la estructura de las páginas de presentación, para la configuración del *Marco*, existen varios *elementos* que coinciden exactamente en ambas clases de equipos. Por ejemplo, el esquema de control de acceso a las páginas de configuración web es común a los dos tipos de equipos principales del sistema.

5.4.2.1 Configuración del Servidor Web en los Equipos Principales (ESCC, ERC)

La versión del software instalada en ambas clases de equipos es la misma. Así, tanto en el ESCC, como en los ERCs se ha instalado la *versión 1.7.4* del software *XAMPP*, el cual, entre otros, incluye las versiones de *software* apropiadas para convertir a los equipos en auténticos servidores web.

Dentro de todo el software que se instala con la aplicación *XAMPP*, los realmente relevantes para la elaboración del *Marco* son:

- *Apache 2.2.17*
- *MySQL 5.5.8*
- *PHP 5.3.5*
- *OpenSSL 1.0.0c*

También se ha hecho uso de algunas aplicaciones como *phpMyAdmin* (*versión 3.3.8*), la cual permite el manejo de las *bases de datos* (*MySQL*) de forma muy cómoda a través de una interfaz web. Esta herramienta se ha utilizado para la resolución de problemas en la fase de desarrollo del sistema.

Una ventaja del software utilizado para convertir los equipos en pequeños servidores web (*XAMPP*) es que la instalación es muy sencilla, ya que lo único que hay que realizar es un desempaquetado de la carpeta y lanzar la aplicación *lampp* al inicio del sistema. La aplicación *lampp* ejecuta como elementos principales las aplicaciones mencionadas anteriormente: *Apache*, *MySQL* y *php*. Para ello, se configura un enlace simbólico (*S90lampp*) al iniciar los sistemas en el *Nivel de Ejecución 2* (*multiusuario con red*). Adicionalmente, se programa el cierre de la aplicación a través de otro enlace simbólico (*K10lampp*), como secuencia de cierre al apagar el equipo. La configuración se realiza como usuario con privilegios.

```
pfc@escc:~$ sudo ln -s /opt/lampp/lampp /etc/rc2.d/S90lampp
pfc@escc:~$ sudo ln -s /opt/lampp/lampp /etc/rc2.d/K10lampp
```

Esta estrategia, como veremos, se utilizará para lanzar otras aplicaciones, necesarias en el proyecto, al iniciar los sistemas.

La aplicación *XAMPP* tiene establecido un método para proveer cierta seguridad a las aplicaciones instaladas. Para activar esta seguridad bajo *Linux*, se ha de ejecutar un *script* e ir introduciendo los parámetros de configuración como es la clave de las diferentes aplicaciones. Esta clave por comodidad la mantenemos idéntica para todas las aplicaciones y accesos:

```
pfc@escc:~$ /opt/lampp/lampp security
XAMPP: Quick security check...
XAMPP: Your XAMPP pages are NOT secured by a password.
XAMPP: Do you want to set a password? [yes] yes
XAMPP: Password:
XAMPP: Password (again):
XAMPP: MySQL is accessible via network.
XAMPP: Normaly that's not recommended.
XAMPP: Do you want me to turn it off? [yes] no
XAMPP: MySQL has to run before I can check the security.
[...]
```

Es necesario mantener el acceso a la *base de datos* desde la red en el *ESCC*, debido a que será éste uno de los métodos por el cual se comprobará, desde los *ERCs*, que el *ESCC* está activo. Sin embargo, este acceso puede ser desactivado en los *ERCs*. También, se modifica el archivo *my.conf* (configuración de *MySQL*) para cuando no haya un servidor *DNS* en la red:

```
[mysqld]
skip-name-resolve
[...]
```

Este parámetro indica a *MySQL*, que no resuelva los nombres de servidor cuando compruebe las conexiones de los clientes. Con esta opción los valores de la columna *host* en la tabla de privilegios deben ser *direcciones IP* o *localhost*[99]. Si no se realiza esta configuración es posible que ciertas respuestas sean lentas cuando no se disponga de un servidor *DNS* en la red.

Se ha de notar que el acceso a la configuración, siempre se realiza desde peticiones *http* a *scripts php* que acceden de forma local a la *base de datos* y no desde acceso directo a la *base de datos*. Esto es debido, en parte, al paradigma de programación implementado, ya que las aplicaciones *swf* no disponen de un driver para interactuar con *bases de datos*; no obstante, comprobando el acceso a la *base de datos* desde la red se asegura que lo tenga el propio equipo y que la aplicación *MySQL* está ejecutándose en el *ESCC* (ver apdo. 5.4.4.4.2 Página de Inicio en el Sistema de Presentación de Contenido (SPC): Pantalla Inicial Sin Conexión).

El usuario que se utiliza cuando se hace uso de las aplicaciones *Apache* desde *XAMPP* es *nobody* dentro del grupo *nogroup*. Esta configuración se puede consultar en */opt/lampp/etc/httpd.conf*, cuando está ejecutándose a través de *XAMPP*.

```
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
user nobody
group nogroup
```

En el archivo de configuración de *Apache*, se recomienda cambiar el *usuario* y el *grupo* a uno específico para el servicio; sin embargo, debido a que este no es un factor que afecta al diseño del *Marco* se mantiene y se le da permisos para que pueda manipular archivos del sistema. La estrategia seguida ha sido preparar el archivo */etc/sudoers* para asignar

privilegios y poder editar ficheros de sistema y ejecutar *script* desde la interfaz web a través de este usuario. Es decir, desde el interfaz web, el usuario administrador de la configuración, aplicará configuraciones que producirá que se editen archivos de configuración del *sistema operativo* y se ejecuten ciertos procesos a través de *script* escritos también en *php*.

Así, debido a que el usuario de *XAMPP* es *nobody* por defecto, aplicamos la posibilidad de ejecutar procesos con privilegios *sudo* desde este usuario. Otra alternativa es definir los permisos en las propiedades de algunos archivos necesarios; no obstante, finalmente se escoge dar privilegios al usuario *nobody* para definir y testear el *Marco*. Las aplicaciones desde el servidor *Apache* pueden ser ejecutadas con introducir la siguiente línea dentro del fichero */etc/sudoers*:

```
nobody ALL=NOPASSWD:ALL
```

Se ha de tener en cuenta que el archivo es de sólo lectura y cada vez que se inicia el *sistema operativo* comprueba que se mantienen estos permisos, por lo que hay que dejar el archivo con los permisos iniciales para que el *sistema operativo* inicie de forma normal:

```
chmod 440 /etc/sudoers
```

Para poder establecer estos permisos se ha de iniciar el sistema en modo seguro (*safe mode*) o alternativamente realizar la edición de */etc/sudoers* con el usuario *root*.

5.4.2.2 Configuraciones Relativas a la Interfaz de Red de lo Equipos Principales (ESCC, ERC) debidas a la Virtualización mediante la aplicación VMWARE

Como se puede consultar en el anexo C Manual del Usuario Administrador, se ha previsto la posibilidad de cambiar la configuración lógica del *protocolo IPv4* de las tarjetas de red a través de la *Interfaz Gráfica de Usuario*. Esta configuración se debe almacenar de forma permanente en el sistema y, para ello, se almacenan los parámetros de configuración en un archivo y en una tabla de la *base de datos* de modo que se aplica la configuración al iniciar el sistema.

En nuestro entorno de trabajo, tal como se ha explicado en el apartado 5.3, se ha realizado la *virtualización* de los equipos principales mediante la aplicación *VMWare* (ver anexo B.I Instalación de Servidor de Máquinas Virtuales. *VMWare Server*). Debido a esta virtualización se han de considerar algunos aspectos en ciertas versiones del *sistema operativo Ubuntu*, lo cual origina una serie de necesidades en la configuración que se debe aplicar al iniciar estos equipos.

Estas acciones se deben incluir dentro de la secuencia de inicio de cada uno de los equipos, siendo idénticas en ambos casos, por lo que se dedica este subapartado para explicar las connotaciones de esta configuración.

Problemas con la Generación de reglas udev en la Virtualización

En las versiones iniciales de *Linux*, cuando el sistema soportaba un tipo de dispositivo, se creaba un nodo en el sistema de archivos que mantenía la información necesaria asociada al driver para así poder manejarlo [100]. Esto se convirtió en un problema cuando el número y tipo de dispositivos creció, ya que habría que crear un sistema de archivos */dev* muy elevado, de los cuales sólo se utilizaban un número pequeño de dispositivos. La solución inicial para los dispositivos en el árbol de directorios */dev* consistía en mantener el mismo nodo en el árbol aunque el dispositivo se desconectara; sin embargo, esto se convirtió en un problema cuando

aparecieron dispositivos que se conectan y desconectan sobre el mismo conector (*USB*) y utilizan drivers diferentes (cámaras, faxes, scanner, ratones,...).

Para solucionar estos problemas apareció *udev*, el cual es un programa que maneja, en el espacio de usuario y no en el *kernel*, todos los dispositivos en el árbol de directorios */dev*, informando, a las aplicaciones de usuario, si el dispositivo está conectado o desconectado. Así, sólo los dispositivos que están conectados al sistema están presentes en el árbol */dev*.

Udev se basa en *sysfs* y utiliza una serie de normas específicas (*udev-specific rules*) para establecer las convenciones de nomenclatura apropiada [100]. *Sysfs* fue introducido en el *kernel* de *Linux* en la versión 2.6. Se trata de un sistema de ficheros virtual (se almacena en memoria y no en disco), que proporciona información detallada y bien organizada sobre los dispositivos disponibles en el sistema, sus configuraciones y su estado. La información de los dispositivos almacenada por *sysfs* es accesible, tanto desde del *kernel*, como desde el espacio de usuario. Así, el árbol de directorios generado por *sysfs* es */sys*, y representa una estructura jerárquica mostrando en cada nodo final un solo atributo del dispositivo subyacente.

Udev genera, en función a un conjunto de reglas y apoyándose en */sys*, la gestión de los dispositivos para que así puedan utilizarlo las aplicaciones. Las **reglas por defecto** residen en el directorio */lib/udev/rules.d* y las **reglas locales** se encuentran */etc/udev/rules.d*. En principio, no hay necesidad de borrar o editar las *reglas por defecto*, ya que se puede ignorar o sobrescribir éstas creando reglas con el mismo nombre en el directorio de reglas locales. Los archivos de reglas se nombran según el patrón *nn-description.rules*, donde *nn* es normalmente un dígito de dos números y los archivos son procesados por orden léxico, de modo que los números inferiores son procesados en primer lugar. El sufijo *.rules* es obligatorio [100].

Además, *udev* gestiona dinámicamente la creación y supresión de ficheros de dispositivos de acuerdo a la presencia real (o ausencia) de estos. El demonio *udev*d escucha los mensajes del núcleo en relación con los cambios de estado del dispositivo. Así, en base a la información de configuración en */etc/udev* y */lib/udev*, *udev*d puede tomar una variedad de acciones cuando un dispositivo se conecta o se desconecta. De forma predeterminada, sólo crea ficheros de dispositivo en */dev* y de esta forma el servicio *udev*d utiliza reglas para establecer el nombre de los dispositivos. Habitualmente las necesidades en los diferentes sistemas son que el *hardware* que normalmente se mantiene estático, mantenga esas características con el mismo nombre y, el que normalmente se conecta y desconecta, establezca una configuración dinámica.

En *Ubuntu*, por defecto se utilizan *scripts* que se ejecutan al inicio del sistema y que generan reglas apropiadas para establecer nombres y enlaces simbólicos apropiados a los dispositivos. La versión 10.10 de *Ubuntu*, utilizando */lib/udev/rules.d/75-persistent-net.rules*, describe las reglas genéricas para asignar los nombres a los dispositivos de red de forma persistente. Esta *regla por defecto*, construida de forma genérica y actualizada con cada versión del sistema operativo, llama a otros *scripts* que generan *reglas locales*. En concreto, en caso necesario, crea la regla en */etc/udev/rules.d/70-persistent-net.rules* para seleccionar los drivers que finalmente se aplican a los dispositivos de red, mediante el *script* *write_net_rules* (ubicado en */lib/udev*).

En versiones anteriores de *Ubuntu* (por ejemplo, en las versiones 9.04 y anteriores) no se consideraba la posible *virtualización*, a la hora de generar los nombres de las interfaces de red de forma persistente. Esto provocaba un problema cuando se copiaba la máquina virtual de un

equipo a otro, debido a que la *dirección MAC* generada de forma automática, por muchas aplicaciones de virtualización, cambia de un *equipo anfitrión* a otro. Así, el principal problema era que cuando se iniciaba el sistema y se disponía de una regla que establecía un nombre (*eth0*) para una determinada *dirección MAC-0*, al detectar otra nueva *dirección MAC-1* en el nuevo sistema anfitrión, el sistema, a través de *udev* proporcionaba otro nombre diferente a la interfaz: *eth1*. El nombre de la interfaz en el sistema iba aumentando en su numeración, de forma progresiva, según se iban realizando copias de las máquinas virtuales.

En *Ubuntu 10.10* el problema se soluciona introduciendo ciertas excepciones al archivo generador de reglas por defecto (`/lib/udev/rules.d/75-persistent-net.rules`) cuando se detecta que la interfaz pertenece a una máquina virtual:

```
# ignore KVM virtual interfaces
ENV{MATCHADDR}=="52:54:00:*", GOTO="persistent_net_generator_end"
# ignore VMWare virtual interfaces
ENV{MATCHADDR}=="00:0c:29:*|00:50:56:*", GOTO="persistent_net_generator_end"
```

Por ejemplo, en una de las máquinas virtuales se generó la siguiente dirección física:

```
eth0      Link encap:Ethernet  direcciónHW 00:0c:29:b3:37:fc
```

Existen otros modos de solucionar el problema, los cuales pueden ser aplicados a versiones anteriores de *Ubuntu 10.10* y que se comentan debido a que han sido testadas sobre las versiones 8.04 y 9.04 de *Ubuntu*:

1. Generar la máquina virtual con una *dirección MAC* estática.
2. Eliminar siempre, al iniciar el sistema (*script php* de inicio), la regla que genera el nombre de la interfaz, de modo que en el siguiente inicio se vuelva a generar una regla para asignar el nombre. De esta forma, se eliminará la persistencia y se nombrará siempre el interfaz de red como *eth0*.

```
sudo rm /etc/udev/rules.d/70-persistent-net.rules
```

Al no haber ninguna *dirección MAC* registrada con un nombre de forma persistente, siempre será la interfaz *eth0* la que se configura en el *ESCC* y los *ERCs* y, por tanto, las máquinas virtuales pueden ser copiadas a diferentes equipos sin variar el identificador de red, el cual será siempre *eth0*.

Activar las Interfaces Necesarias de Red

Al iniciar el sistema es necesario activar las interfaces de red para así aplicarles la configuración almacenada en `/etc/network/interfaces`. Estas configuraciones se incluyen dentro de los *scripts php* de inicio de cada equipo.

```
ifup eth0
ifup eth0:0
```

Como en principio se espera que los equipos dispongan de una única tarjeta de red, siempre se configurará en el identificador *eth0*. De este modo activar las tarjetas con el comando *ifup*, se provee la configuración almacenada en el archivo:

```
/etc/network/interfaces
```

La configuración de `/etc/network/interfaces` se puede editar a través de la *interfaz gráfica web* desarrollada, por lo que una vez modificado algún parámetro se editará

dicho fichero. Como se observa, también se activa la sub-interfaz *eth0:0* como estrategia para mantener una forma de conexión permanente, independientemente de la *dirección IPv4* asignada a la interfaz principal *eth0*. Así, se asigna una *dirección IPv4* fija (169.254.69.13) a todos los equipos de forma estática en la sub-interfaz *eth0:0*.

```
pfc@esc: $ cat /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet static
    address 192.168.1.7
    netmask 255.255.255.0
    gateway 192.168.1.1
iface eth0:0 inet static
    address 169.254.69.2
    netmask 255.255.255.0
```

Esta configuración es muy útil para el usuario final, por si el administrador no recordara la *dirección IPv4* asignada al equipo en configuraciones anteriores de modo que no podría acceder a la *Interfaz Gráfica de Usuario* de la cual hace uso el *Administración del Sistema*. El sistema *Linux* no envía paquetes por esta red a no ser que existan otros dispositivos en esa misma subred que pretendan conectarse con el equipo. Esto es debido a que no se establece ningún *gateway* en la configuración.

Esta configuración se ha llevado a cabo, por tanto, para tener accesibilidad al equipo directamente a través de un cable cruzado desde otro equipo; es decir, no es recomendable acceder a través de una *LAN* debido a las incongruencias que pudieran establecerse en el protocolo *ARP* si hubiera más de un equipo en la misma *LAN*. Esto no es en ningún caso crítico debido a que el objeto es la interconexión física de un equipo a otro y no se utilizaría este direccionamiento a través de la interconexión de la *LAN*.

Por tanto, la primera vez que se accede al equipos se puede acceder a través de un cable cruzado directamente conectado al equipo y acceder a través del navegador web a la página de administración del equipo (*dirección IPv4* permanentemente configurada en el interfaz *eth0:0*).

5.4.2.3 *Aplicación Web para Proporcionar una Interfaz Gráfica de Usuario en los Equipos Principales (ESCC, ERC)*

Se ha proporcionado una *Interfaz Gráfica de Usuario* amigable al administrador del sistema para que éste sea capaz de ajustar la configuración. La *Interfaz Gráfica de Usuario* se ha desarrollado sobre el *Marco* diseñado para establecer la configuración (apdo. 5.2.1). En este sub-apartado se explica cómo se ha desarrollado la *Interfaz Gráfica de Usuario* para que el administrador pueda realizar las configuraciones oportunas en los equipos principales.

Con el *Marco* desarrollado para establecer la configuración, se podrá establecer el funcionamiento de la arquitectura para que se adapte a las necesidades de la aplicación de *Difusión Selectiva*. De esta forma un administrador del sistema, no tiene que conocer cómo han sido estructurados los datos internamente, ni tampoco tiene por qué conocer las tecnologías con las cuales se ha desarrollado el *Marco de Aplicaciones de Difusión Selectiva*. El manual de la *Interfaz Gráfica de Usuario* se puede consultar en el ANEXO C Manual del Usuario Administrador.

5.4.2.3.1 Secuencia Lógica de Funcionamiento de la Aplicación Web de Configuración (Interfaz Gráfica de Usuario)

La aplicación web de configuración está formada por *scripts php* que denominaremos **archivos php base**. Estos archivos se generan mediante un mismo patrón y siguen una secuencia lógica que es utilizada en el *Marco* para establecer la configuración tanto en el *ESCC* como en un *ERC*. Aunque no todos los elementos en la secuencia se utilizan en ambos equipos, se describen en un apartado común debido a que los *archivos php base* siguen un mismo esquema.

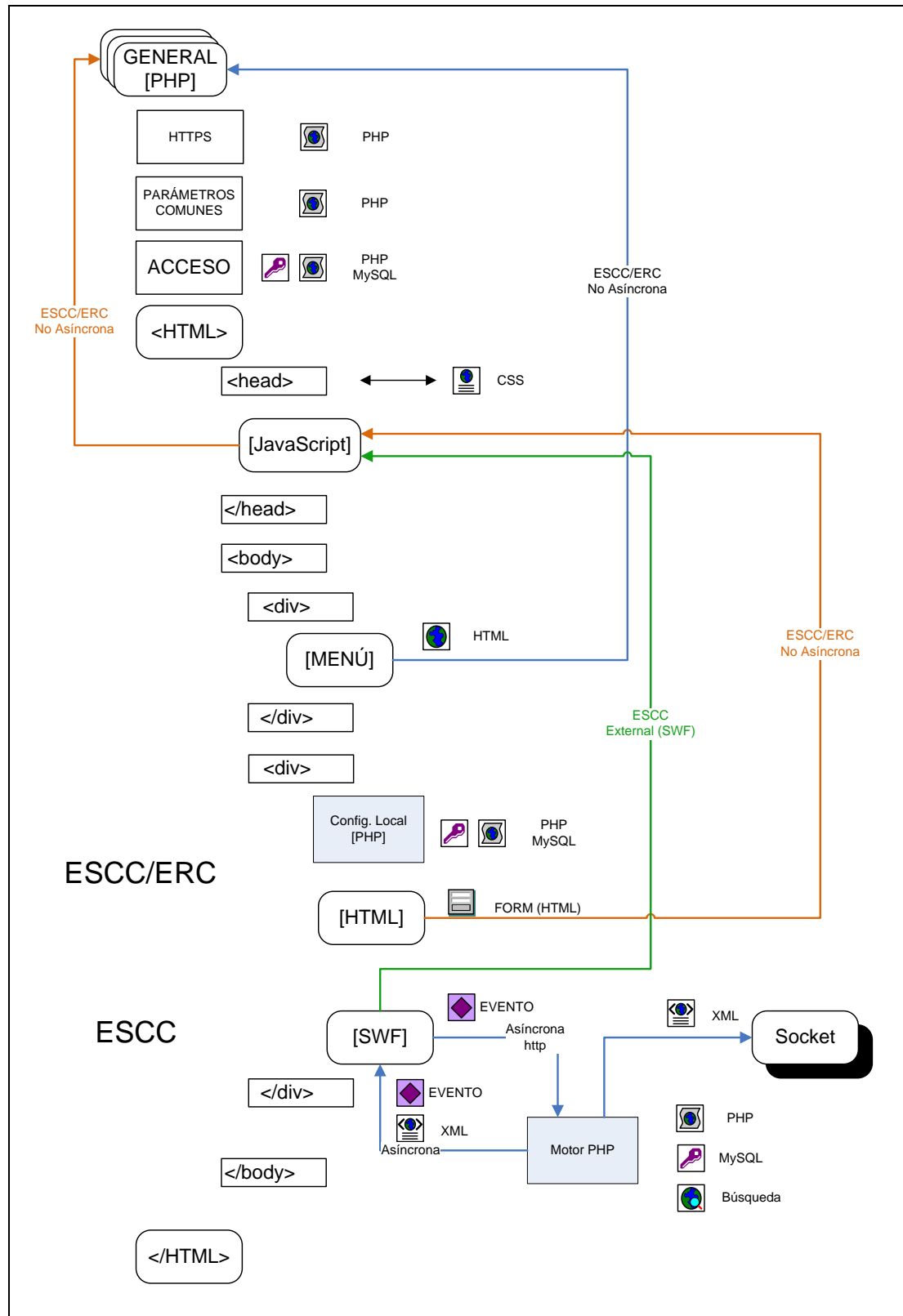


Ilustración 35.-. Secuencia lógica de los archivos programados en lenguaje *php* correspondientes a la interfaz de usuario del administrador del sistema: archivos *php* base.

La Ilustración 35 muestra, de forma esquemática, la lógica de los *archivos php base* que permiten el acceso a la configuración del sistema a través de un navegador web. Está

íntimamente ligada con el diseño del *Marco* (ver apartado 5.2.1 Marco Diseñado y Desarrollado para Establecer la Configuración).

Los *archivos php base (scripts php)* devuelven principalmente código *html* y facilitan la configuración de la aplicación al administrador del sistema. Por tanto, la Ilustración 35 muestra la estructura de estos archivos, los cuales son la *base* o soporte de la *interfaz de usuario: código html con objetos swf embebidos y funciones JavaScript definidas*. En la ilustración se muestra también algunas interacciones con elementos de importancia en los equipos (*Motor PHP, Servidor Socket*).

De esta forma, los objetos *swf* embebidos permiten al administrador del sistema la interacción *amigable, asíncrona y eficiente* con el servidor. Las *aplicaciones swf* establecen comunicación con los *scripts php (Motor PHP)*, los cuales acceden a las *bases de datos* y los archivos del *sistema operativo*. Las funciones *JavaScript* adquieren el control de la aplicación en el lado del navegador del administrador del sistema (*EACCC*), cuando se realizan determinadas acciones; desde la aplicación *swf*, por ejemplo, se llaman a estas funciones (*JavaScript*) mediante la clase *ExternalInterface* cuando se producen determinados eventos registrados en cada aplicación *swf*.

Los **archivos php base** no son parte del *Motor PHP* (ver apdo. 5.4.3.4.1.2 Motor PHP en el *ESCC*), ni tampoco son parte de la *Configuración Local* que se programa también en *php*. Las peculiaridades del *Motor PHP* y los *scripts* de *Configuración Local* de cada tipo de equipo (*ESCC* y *ERCs*) las veremos en posteriores subapartados. Como se adelanta en la Ilustración 35, el *Motor PHP* y la *Configuración Local* son archivos escritos en *php*, pero programados para realizar acciones muy concretas y en la mayoría de los casos interactuando con la configuración almacenada en la *base de datos*.

La secuencia lógica que describe los **archivos php base**, que se muestra en la Ilustración 35, representa una estructura muy genérica y, como se ha explicado, estos archivos serán los responsables de mostrar la composición de la *Interfaz Gráfica del Usuario*. Esta *Interfaz Gráfica de Usuario* presentará la configuración establecida hacia el administrador del sistema.

A continuación, en los sucesivos sub apartados, se hace un breve resumen de los elementos más significativos, a los cuales denominamos módulos.

5.4.2.3.1.1 MÓDULO HTTPS

Este módulo fuerza que la conexión se realice a través del protocolo *https*.

Cuando se hace una petición a una página determinada del *Marco* se comprueba, mediante código *php*, si se ha realizado dicha petición mediante el protocolo *https*. Si es así, se sigue cargando la página y si no, se redirige el tráfico a la misma *URL* utilizando seguridad *https* y estableciendo una conexión segura.

```
$uri = 'https://';
$uri .= $_SERVER['HTTP_HOST'];

if (empty($_SERVER['HTTPS']) || ('on' != $_SERVER['HTTPS'])) {
    header('Location: '.$uri);
}
```

Forzar al protocolo *https* es una medida en la que se solventan algunas cuestiones de seguridad y ayuda a hacer el sistema más robusto frente a ataques, sobre todo al acceder a las páginas de configuración donde se intercambian en texto en claro las claves para acceder al sistema en la pantalla de acceso.

5.4.2.3.1.2 **MÓDULO DE PARÁMETROS COMUNES**

Existen ciertos parámetros comunes que se incluyen al inicio de los *archivos php base*. Estos parámetros son comunes a todos los *archivos php base* y al tipo de equipo que estemos tratando; es decir, incluiremos ciertos parámetros en los *archivos php base* del *ESCC* y otros diferentes en los *archivos php base* de los *ERCs*.

Los archivos que se incluye también están escritos en *php* debido a que es necesario incluir cierto dinamismo y estos valores se almacenan en variables que son utilizadas en otros segmentos al generar el código *html*.

Los parámetros comunes estáticos más importantes son:

- versión del software y por tanto la estructura de la *base de datos* (establecido ante posibles evolución del *Marco*).
- el *nombre de la base de datos* en la cual se deben realizar las operaciones
- puertos utilizados en las comunicaciones
 - puerto para realizar las peticiones *http* (TCP 8080)
 - puerto del *servidor socket* (TCP 8081)
- el usuario y la contraseña que permiten el acceso a la *base de datos*

5.4.2.3.1.3 **MODULO DE ACCESO**

Se ha tenido en cuenta un *control de acceso* para la configuración del sistema.

Este módulo se identifica dentro de la Ilustración 35 con el recuadro etiquetado como 'ACCESO'. Este subconjunto de archivos es necesario debido a que el acceso se realiza a través de un navegador web que hace peticiones a un servidor cuyo acceso no está restringido por el propio servidor web (*Apache*). El acceso dentro de la *arquitectura funcional* y, por tanto, del *Marco*, se ha previsto que se establezca a través de una red mediante varios usuarios. Por tanto, debe ser la propia *Interfaz Gráfica de Usuario* la que limita el acceso a las distintas páginas de configuración. El mecanismo de control de acceso está compuesto por varios archivos que proveen al sistema de cierta seguridad, de esta forma el módulo restringe el acceso a la configuración a determinados usuarios administradores del sistema.

El *control de acceso* ideado se utiliza en cada unas de las páginas de configuración de acceso, tanto en el *Equipo Servidor de Configuración y Contenido (ESCC)*, como al acceder a la configuración del *Equipos Reproductores Clientes (ERC)*.

Antes de aplicar la lógica de acceso se ha programado en las páginas el MÓDULO HTTPS, para que se realicen todas las peticiones mediante el protocolo *https* (apartado 5.4.2.3.1.1). De este modo, se añade cierta seguridad a la configuración del sistema ya que los *usuarios* y las *claves* introducidas en el formulario se enviarán cifrados por la red cuando se esté intentando acceder.

El flujo lógico del sistema de control de acceso diseñado y desarrollado se muestra de forma gráfica en el diagrama de la Ilustración 36.

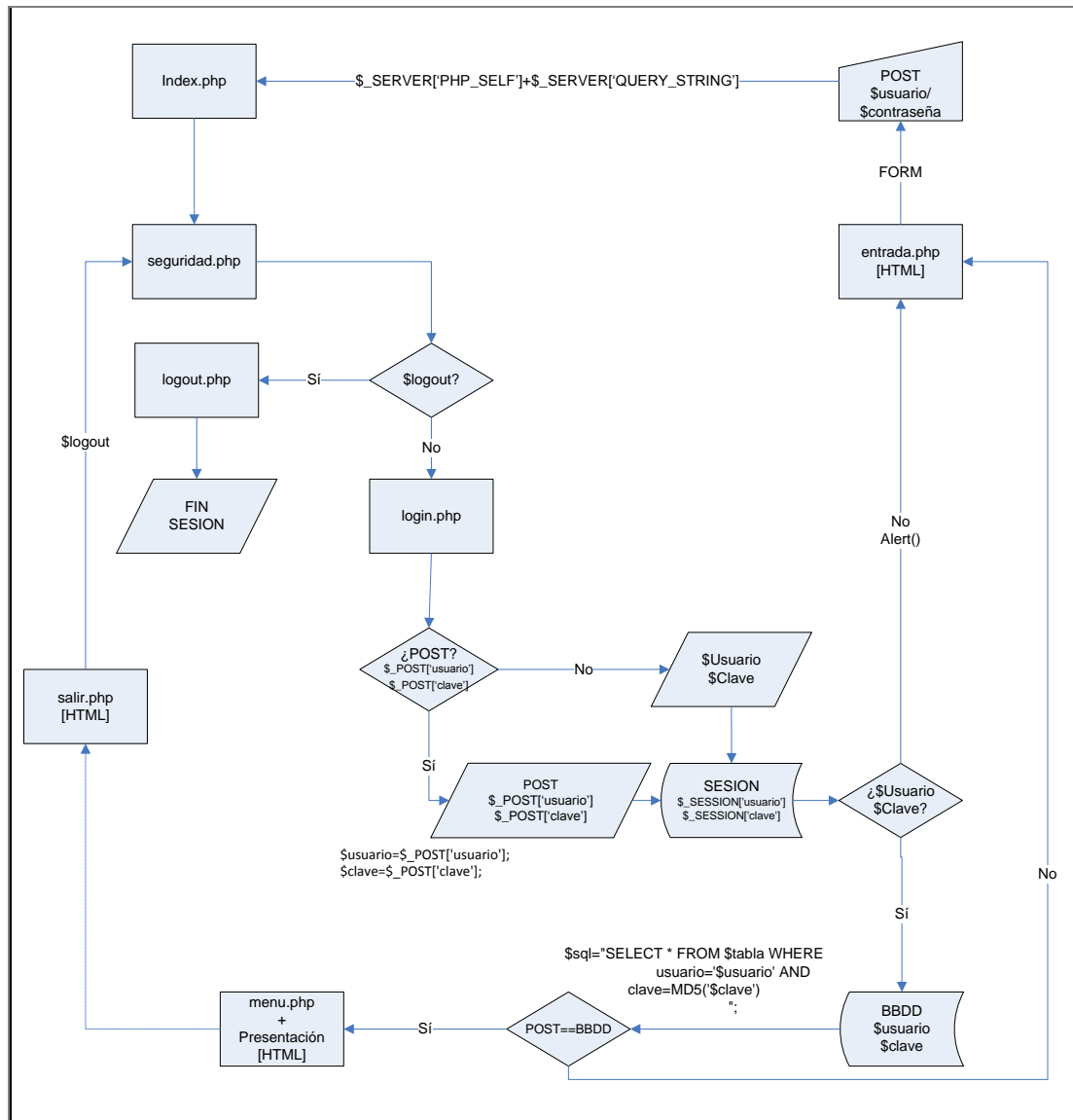


Ilustración 36.-. Resumen del flujo lógico del bloque que restringe el acceso a la configuración del sistema.

Las tablas de la *base de datos (MySQL)* almacenan los datos de acceso de forma local. Los usuarios ('*admin*' y '*sistema*') y las contraseñas de acceso se almacenan como resumen MD5 en cada una de las *bases de datos locales*²⁰. La razón de esto es la configuración independiente de los distintos equipos. En la Ilustración 36 se hace un resumen de diagrama de flujo de la verificación del acceso.

Al acceder a cada página principal que permite la configuración del equipo se invoca un *script* escrito en *php* denominado *index.php*. Este *script* muestra la página de inicio si el proceso de validación de acceso se supera. Este *script php*, al igual que el resto de páginas, incluye el *script* que provee la seguridad en el acceso a los archivos de la aplicación: *seguridad.php*.

²⁰ Cada equipo ya sea *ERC* o *ESCC* almacena una *base de datos* de forma local con la configuración del equipo. En esta configuración se almacenan los usuarios a los que se les permite el acceso y las contraseñas como resumen MD5.

El *script* `seguridad.php`, analiza si se ha introducido `$logout` (se ha pulsado salir o si existe la variable). Si la acción es verdadera se carga el *script* `logout.php` que lo que hace es simplemente finalizar la sesión y muestra código *html* indicando que la sesión ha finalizado.

En el caso de que no se haya salido de la sesión se carga el *script* `login.php`. En éste se evalúan una serie de condiciones para reconocer si el *usuario* y la *clave* que se han introducido coinciden con los configurados en el sistema:

- Se evalúa si se ha enviado mediante el método *POST*, a través de un formulario escrito en *html*, las variables `$_POST['usuario']` y `$_POST['clave']`.

- En caso afirmativo, se utilizarían el par enviado como variables de sesión, además de almacenar el valor enviado en las variables `$usuario` y `$clave`.

```
$usuario=$_POST['usuario'];
$clave=$_POST['clave'];
$_SESSION['usuario']=$usuario;
$_SESSION['clave']=$clave;
```

- En caso negativo, se utilizarían los valores almacenados en `$usuario` y `$clave` para establecer la sesión, ya que se habrán establecido con anterioridad.

```
$usuario=$_SESSION['usuario'];
$clave=$_SESSION['clave'];
```

- Posteriormente, se evalúan las variables `$usuario` y `$clave` comprobando si coinciden con los valores almacenados en la *base de datos*, previo resumen en MD5 de la variable `$clave`.

- Si se verifica la coincidencia, se presenta la página en cuestión (en principio `index.php`). Esta página incluye el **menú** y el código en *html* que muestra el contenido de la página, el cual será siempre un *archivo php* *base* similar al presentado en la Ilustración 35.
- En caso de que la validación no se produzca debido a que no se produjera la coincidencia, se muestra un mensaje de error y se presenta de nuevo un *formulario* en el que se pide el *usuario* y la contraseña o *clave* (`entrada.php`). El mecanismo se implementa bajo peticiones *https*, así en el formulario no se envía la clave en texto en claro al intentar acceder.

- `$_SERVER['PHP_SELF']` devuelve la ruta absoluta al *script* del *php* en ejecución, esto es importante al crear el formulario, porque así se puede enviar a `index.php` (u otra página cualquiera) que es la URL del *php* en ejecución. De esta forma no es necesario conocer el dominio o la *dirección IPv4* y no es necesario conocer la ruta absoluta.
- `$_SERVER['QUERY_STRING']` devuelve los parámetros que se envían en la URL.

La validación del usuario y la clave podría realizarse de varias formas. Una alternativa a almacenar las claves de forma local en la *base de datos* de cada equipo es la de realizar la validación de forma remota mediante cualquier otro mecanismo. Por ejemplo, el acceso de determinados usuarios podría validarse siempre contra el ESCC, de modo que la clave de administración del sistema se mantuviese centralizada incluso para la configuración de los ERCs; sin embargo, al menos habría que definir un usuario de forma local para tener acceso a

la modificación de los parámetros de red del equipo, de forma que se pudiera configurar la conectividad previamente. Otras alternativas de validación pueden ser mecanismos más sofisticados, como *LDAP* (*Lightweight Directory Access Protocol*, *Protocolo Ligero de Acceso a Directorios*) o cualquier otro protocolo de validación.

El mecanismo de acceso desarrollado resulta muy modular debido a que se incluye, en cada archivo que compone la *Interfaz Gráfica de Usuario*, el archivo *seguridad.php* (mediante la directiva *include()*) manteniendo y verificando la sesión con el usuario. En caso de que se modifique el mecanismo de *control de acceso* (por ejemplo, realizando una validación con un servidor *LDAP*), tan sólo hay que modificar los archivos involucrados en el control de acceso; sin embargo, no es necesario modificar ninguno del resto de archivos donde se presentan la interfaz de usuario, siendo esto una ventaja importante ante mejoras futuras de la aplicación.

Este esquema se implementa cada vez que navegamos por las distintas páginas de configuración, tanto del *ESCC*, como del *ERC*. La excepción es el archivo *index.php principal* (no el *index.php* del MÓDULO ACCESO²¹). El archivo *index.php principal* es básicamente una división de *frames html* donde se inserta la aplicación web final.

5.4.2.3.1.4 MÓDULO JAVASCRIPT

Se definen e incluyen funciones *JavaScript* que obtienen el control de la ejecución en el navegador del administrador (*EACCC*) en determinadas situaciones. La inclusión de funciones *JavaScript* se muestra en la Ilustración 35 mediante el módulo etiquetado como [*JavaScript*]. Estas funciones son diferentes dependiendo del tipo de archivo que se está ejecutando en el servidor.

Algunas de las situaciones más trascendentes donde el navegador adquiere el control de la ejecución son las siguientes:

- Cuando se envía un formulario, se comprueba que los datos introducidos en ciertos campos son adecuados.
- Cuando se produce un determinado evento, en un *aplicación swf* que se está ejecutando en el navegador, se invocan las funciones *JavaScript* que se han definido para la página mediante la clase *ExternalInterface* (clase *ActionScript 3*), desde la propia aplicación *swf*. Las funciones *JavaScript* que se han escrito pueden mostrar mensajes de advertencia o confirmar determinadas acciones, incluso pueden redirigir la exploración hacia otra *URL*.

Como se puede apreciar, *JavaScript* nos proporciona un mecanismo muy sencillo para establecer nuevas peticiones al *ESCC* desde el *ERC* o el *EACCC*. Así, según los eventos producidos dentro de las aplicaciones *swf* se puede llamar a una u otra función *JavaScript*. Esta característica se convierte en una herramienta clave a la hora de diseñar y desarrollar el *Marco para Aplicaciones de Difusión Selectiva*, tanto para establecer la configuración, como para proporcionar el funcionamiento como veremos (apdo. 5.4.4.3.2 Aplicación Control swf y Clientes Socket en los ERCs).

²¹ El archivo *index.php* del MÓDULO DE ACCESO sí hace uso del mecanismo de control de acceso, *include(seguridad.php)*, ya que será la página principal a partir de la cual el usuario entra en el sistema por primera vez para una determinada sesión.

5.4.2.3.1.5 MÓDULO MENU

El módulo, etiquetado en la Ilustración 35 como [MENÚ], consta de los enlaces a las opciones de configuración del equipo que se esté configurando: *ESCC* o *ERC*. Se ha escrito en *php*, porque se necesitan recuperar variables dependiendo de la página a la que se quiere acceder, por lo que existe cierto dinamismo. Se introduce un menú (a modo de lista aplicando un *CSS*) común a las páginas de administración del sistema. Este módulo se carga en los *archivos php base* y nos permite navegar entre cada una de las páginas de configuración estableciendo una *Interfaz Gráfica de Usuario*.

Se han definido dos **niveles de administración o de acceso** tanto en el *ESCC*, como en los *ERCs*. Estos usuarios se han denominado 'sistema' y 'admin'.

El usuario **admin** tiene acceso a los elementos del menú relacionados con la configuración de la *Aplicación de Difusión Selectiva* y, por otra parte, el usuario **sistema** tiene acceso, además, a los elementos que cambian parámetros del *sistema operativo*:

- **sistema:** Accede a todos los elementos posibles de configuración del menú. Incluye el acceso a los formularios relativos a la configuración del *sistema operativo* como son las *direcciones IPv4* del equipo o los servidores de hora (referencias *NTP*) en el *ESCC*, de los cuales se adquirirá la referencia temporal para sincronizar el resto de equipos. Además del acceso a estas configuraciones, también está habilitado para configurar los parámetros relacionados con la *Aplicación de Difusión Selectiva*: *ubicaciones, listas programadas, archivos, zonas,...*
- **admin:** Este usuario tiene restringidos el acceso a las páginas de configuración del *sistema operativo*, así mostrará la posibilidad de acceder a los elementos que estén relacionados con la administración de la *Aplicación de Difusión Selectiva*.

Se programan, dentro de los *archivos php base*, dos consideraciones para definir los permisos de estos usuarios:

- Dentro del módulo MENÚ, si el usuario con el que se ha accedido (`$_SESSION['usuario']`) coincide con el usuario *sistema* se añade al menú la posibilidad de acceder al enlace *del archivo php base* relativo a esa configuración.
- Al acceder a la página en concreto se verifica que el acceso está permitido para el usuario (`$_SESSION['usuario']`).
 - Si está permitido, se sigue la carga de la página.
 - Si no, se borra la sesión y se redirige a la página *salir.php* perteneciente al control de acceso del sistema, de modo que el comportamiento es el mismo que si se hubiera salido del sistema por voluntad propia.

A continuación (Tabla 12), se presenta una tabla con los enlaces y páginas de acceso a las que se ha permitido a cada uno de los usuarios. Para el ESCC:

ESCC				
MENU	archivo php base	Descripción	SISTEMA	ADMIN
INICIO	Index.php	Presentación al acceder al sistema.	✓	✓
UBICACIONES			✓	✓
Seleccionar/Crear Ubicación	Ubic.php	Seleccionar y Crear <i>ubicaciones</i> . Según la <i>Ubicación</i> que se seleccione en este apartado se estarán modificando parámetros de ésta en el resto del acceso al menú.	✓	✓
Configurar Ubicación				
Zonas a Esc	zonasEsc.php	<i>Zonas a Esc</i> . Se dibujan las <i>Zonas</i> en el <i>Escenario</i> . Se permiten hasta 4 zonas.	✓	✓
Listas a Zonas	listasZonas.php	<i>Listas a Zonas</i> . Se asignan las listas programadas a cada zona según una fecha.	✓	✓
Archivos a Listas	archivosListas.php	<i>Archivos a Listas</i> . Se programan los archivos en intervalos en cada lista de reproducción de la <i>Ubicación</i> .	✓	✓
Equipos a ubicaciones				
Equipos ^a ubicaciones	equiposUbic.php	<i>Equipos a ubicaciones</i> . Se configuran los equipos que estarán asignados a cada <i>Ubicación</i> . La programación se realizará en función de la <i>dirección IPv4</i> .	✓	✓
ARCHIVOS			✓	✓
Gestión de Archivos	gestionArchivos.php	Subir y borrar archivos. Permite subir y archivos multimedia (imágenes y videos) al ESCC	✓	✓
Búsqueda de Archivos	buscaArchivos.php	Permite buscar archivos dentro de las programaciones e identificar a que listas y Ubicación está asignado.	✓	✓
ACTUALIZAR			✓	✓
Actualizar contenido	actualizaCont.php	Actualizar Contenidos en <i>Ubicación</i> . Emite la orden de actualización del contenido a una determinada <i>Ubicación</i> .	✓	✓
Actualizar Ubiq/Equipos	actualizaPersonalizado.php	Actualizar ubicaciones o equipos. Permite seleccionar el tipo de actualización: de presentación o de sincronismo de archivos; a una determinada <i>Ubicación</i> o a un determinado ERC	✓	✓
SISTEMA			✓	✗
Configuración Servidor				
Parámetros de Red	IPServidor.php	Permite modificar los parámetros de red del ESCC	✓	✗
Servidores DNS	DNSServidor.php	Permite configurar la <i>dirección IPv4</i> del servidor DNS.	✓	✗
Referencias NTP	NTPServidor.php	Permite configurar la referencia NTP que establecerá el ESCC para ajustar la hora.	✓	✗
Cambia Claves	cambiaContra.php	Cambia las claves de los usuarios.	✓	✗
Estado Servidor				
Estado de Red	StatusCambiIP.php	Muestra el estado de la conectividad IPv4	✓	✗
Estado DNS	StatusDNS.php	Muestra el estado del servidor DNS.	✓	✗

ESCC				
MENU	archivo php base	Descripción	SISTEMA	ADMIN
Estado NTP	StatusNTP.php	Muestra el estado de conectividad con el servidor NTP además de la hora establecida.	✓	✗
Reiniciar				
Reiniciar Servidor	reiniciarServidor.php	Permite reiniciar el servidor de forma remota.	✓	✗
DESCARGAS			✓	✓
Enlaces Software				
Windows	Descarga_Software.php? SO=windows	Permite acceder a los enlaces de descarga de software complementario para la Aplicación de Difusión Selectiva bajo el <i>sistema operativo</i> Windows.	✓	✓
Linux	Descarga_Software.php? SO=linux	Permite acceder a los enlaces de descarga de software complementario para la Aplicación de Difusión Selectiva bajo el <i>sistema operativo</i> Linux.	✓	✓
Archivos				
Imágenes	ListadoArchivos.php? imag=imag	Lista los archivos de tipo imágenes que se han cargado en el servidor.	✓	✓
Videos	ListadoArchivos.php? video=video	Lista los archivos de tipo video que se han cargado en el servidor.	✓	✓
Todo	ListadoArchivos.php? imag=imag&video=video	Lista los archivos multimedia que se han cargado en el servidor.	✓	✓
BackUP ²²	backup.php	Permite realizar y restaurar una copia de seguridad del sistema.	✗	✗
MANUAL			✓	✓
wiki	mediaWiki.php	Media Wiki preparada para ser editada.	✓	✓
SALIR	salir.php	Cierra la sesión	✓	✓

Tabla 12.-. Resumen de archivos php base en el ESCC y relación de acceso desde cada usuario definido en la aplicación de Difusión Selectiva desarrollada.

Para los ERCs, se tiene el siguiente menú según el usuario con el que se accede:

ERC				
MENU	archivo php base	Descripción	SISTEMA	ADMIN
INICIO	Index.php	Página de presentación al acceder mediante validación.	✓	✓
CONFIGURACION			✓	✓
Direccionamiento IP ESCC	IPServ.php	Dirección IPv4 del ESCC al que realiza las peticiones de archivos y configuración. <i>Servidor socket</i> .	✓	✓
Ancho de Banda Sincro.	AnchoBandaSincro.php	Ancho de banda con el que realiza la descarga de archivos desde el ESCC para almacenarlos de forma local.	✓	✓
SISTEMA			✓	✓
Direccionamiento IP ERC	IPCliente.php	Configuración del direccionamiento lógico del ERC.	✓	✗
Clave Admin Equipo	CambiaPassAdmin.php	Cambio de Clave del usuario 'admin'.	✓	✗
Clave Sistema	CambiaPassSistema.php	Cambio de clave del usuario 'sistema'	✓	✗
Reiniciar	reiniciarCliente.php	Permite reiniciar el ERC de forma remota.	✓	✗
ESTADO			✓	✓
Estado Red Cliente	StatusCambioIP.php	Estado de la conexión IPv4 con los elementos que intervienen en la	✓	✓

²² Funcionalidad no implementada en ninguno de los usuarios. Se ha previsto e implementado la configuración de un servidor de Backup; sin embargo, se deja como línea de trabajo futuro.

ERC				
MENU	archivo php base	Descripción	SISTEMA	ADMIN
		arquitectura.		
PING Servidores	PINGServ.php	PING a los servidores ESCC y Backup	✓	✓
Archivos Locales	ListadoArchivos.php	Lista los archivos multimedia almacenados localmente en el ERC.	✓	✓
SALIR	salir.php	Cierra la sesión.	✓	✓

Tabla 13.-. Resumen de archivos php base en el ERC y relación de acceso desde cada usuario definido en la aplicación de Difusión Selectiva desarrollada.

En el anexo C (Manual del Usuario Administrador), se puede comprobar gráficamente cuál ha sido el resultado de la implementación de cada *archivo php base* que se lista en las tablas anteriores. En ese anexo se muestra también la forma de configurar cada uno de los parámetros que se mencionan en la tabla de forma explícita.

Como se puede apreciar de las tablas, es evidente que un *mecanismo de usuarios multinivel* se puede implementar de este modo. De esta forma cada usuario puede acceder a distintas páginas de configuración según los permisos asignados. Se podría programar una página de configuración de usuarios en la que en una tabla se marquen las páginas a las que se tendrá acceso, aunque se deja como característica para añadir en el futuro (ver apdo. 10.2 Líneas de Trabajo Futuro).

5.4.2.3.1.6 MÓDULO HTML Y Config. LOCAL

Este módulo, el cual se presenta también en la Ilustración 35, se utiliza en los *archivos php base* de la *Interfaz Gráfica de Usuario* en los que se están configurando parámetros correspondientes al *sistema operativo* (*dirección IPv4, máscara de red, gateway por defecto,...*) o parámetros de aplicaciones terceras (*ancho de banda para sincronización de contenido, servidores NTP*). Es decir, este módulo permite configurar los parámetros locales de los distintos equipos en el que se apoya el sistema de *Aplicaciones de Difusión Selectiva* para componer el *Marco*.

En el *ESCC* este módulo sólo está presente en los *archivos php base* que pueden ser accedidos en exclusiva por el usuario *sistema* ya que es utilizado únicamente para la configuración de parámetros del *sistema operativo* y las aplicaciones de terceros (*ntp, rsync*).

En los *ERCs* este módulo es el que permite la configuración de todos los parámetros del equipo, ya que el objetivo de la configuración de parámetros sobre el *ERC* es el de obtener conectividad con el *ESCC*.

Entre estos parámetros configurables tenemos:

- las *direcciones IPv4, máscara de red y router por defecto*
- referencias y *servidores NTP*
- servidores *DNS* en el *ESCC*
- servidor de referencia *ESCC* en el *ERC*
- cambio de contraseñas en el *ERC*

Los parámetros se configuran a través de formularios que al ser enviados ejecutan *scripts* en los equipos principales (*ESCC, ERC*) y que hacen llamadas internas a determinadas aplicaciones que configuran finalmente de forma explícita estos parámetros. Los parámetros configurados por estas aplicaciones se almacenan, también, de forma persistente en la *base de datos* del equipo en cuestión.

5.4.2.3.1.7 **MÓDULO SWF**

Este módulo es el más importante de todos los presentados de cara a la definición de una *Interfaz Gráfica de Usuario* en el *ESCC*. Este módulo no se utiliza en el *ERC*.

El módulo está formado por un conjunto de *aplicaciones swf* que se incluyen como objetos de forma embebida en los *archivos php base* y que se ejecutan mediante el *plugin Flash Player* en el navegador. Estas aplicaciones han sido generadas a partir de la combinación de código *ActionScript 3* y componentes de *Flash Player*, compilados con *Adobe Flash Professional CS3*²³.

La característica singular de estas aplicaciones es que manipulan videos e imágenes, y permiten intercambio de datos *asíncrono* debido a que los archivos pueden consumir importantes cantidades de memoria y *ancho de banda*. Además, estas aplicaciones también son de gran importancia dentro del *Marco* debido a que marcan, a través de la interacción con el usuario, eventos en el sistema. La importancia radica en que estos eventos generan órdenes (al *ESCC*) de escritura sobre el *servidor socket*. Estas órdenes que finalmente aplica el *ESCC* sobre el *servidor socket*, producen nuevos eventos en los *ERCs* haciendo que éstos, por tanto, realicen determinadas acciones que afectan al funcionamiento del sistema en su conjunto.

Se ha programado que muchas de las aplicaciones *swf* que forman el módulo monitorizan y muestran, mediante conexiones a este mismo *socket* y desde la *Interfaz Gráfica del Usuario* administrador, información relevante del estado del sistema que puede ayudar a establecer diagnósticos.

La lógica general de las aplicaciones *swf* que forman parte del módulo, se detalla en el apartado 5.4.3.4.1.1 Esquema General de ActionScript: Aplicaciones SWF en el *ESCC*.

5.4.2.3.1.8 **MÓDULO SOCKET**

Se ha programado un *servidor socket* para enviar las órdenes a los *ERCs* desde el *ESCC*. Este elemento es tratado en detalle en el apartado 5.4.3.4.2 Servidor Socket en el *ESCC*, ya que tiene implicaciones importantes en el desarrollo del *Marco*; sin embargo, a continuación resumimos parte del funcionamiento de éste por ser parte de los *archivos php base* debido a la fuerte interacción con otros módulos.

El servidor de *socket* se ejecuta en el servidor a través de un *script php* que es el encargado de gestionar las conexiones y que se ejecuta al iniciar el *ESCC*.

La escritura en el *socket* se hace a través de un *script php* que se ejecuta exclusivamente en el *ESCC* y que es lanzado a petición de las aplicaciones *swf* desde las que se está realizando la configuración del sistema, como se ha mencionado en el subapartado anterior (5.4.2.3.1.7 MÓDULO SWF). La escritura se realiza cuando se produce un cambio en la configuración del sistema que requiere que los *ERCs* realicen alguna acción; por ejemplo, recargar la configuración de la *Ubicación* en la que está configurado el equipo o iniciar la sincronización de archivos multimedia con el *ESCC*.

Cuando se realiza una escritura en el *servidor socket* (la cual se establece en formato *XML*) se reenvía esta información a todos los *ERCs* que tienen establecida la conexión con ese *socket* y que deben ser todos los *ERCs* que estén activos.

²³ Existen nuevas versiones de la aplicación *Adobe Flash Professional CS3*: *CS4* y *CS5*; sin embargo, el lenguaje *ActionScript 3* no ha sufrido modificaciones y las mejoras aportadas parecen que no influyen en el desarrollo del proyecto.

La conexión y la lectura del *socket* se realizan desde los *ERCs* a través de una aplicación *swf* que está embebida como objeto (*control socket*) en la página web que presenta los vídeos e imágenes, cuyo funcionamiento se detalla en el apartado 5.4.4.4.3.2 Aplicación Control *swf* y Clientes *Socket* en los *ERCs*. De esta forma, al interpretar los datos escritos en el *servidor socket*, el *ERC* realizará la acción oportuna diferenciando si esos datos van dirigidos a él o a otro/s equipo/s.

Adicionalmente, algunas aplicaciones que se corresponden con el *módulo swf* también realizan una conexión al *socket* para monitorizar el sistema, por lo que también se realizan lecturas del *socket* desde el *EACCC*.

5.4.2.3.1.9 MÓDULO DE MOTOR PHP

La misión principal del *Motor PHP* es la de configurar las *bases de datos* a partir de las peticiones *http* y parámetros (método *GET* o *POST*) enviados desde las aplicaciones *swf*. Estas *aplicaciones swf* se ejecutan en el navegador del administrador del sistema en el equipo *EACCC* y realizan llamadas *asíncronas* al *Motor PHP*. Además el *Motor PHP* controla algunos aspectos asociados a la subida de archivos multimedia al servidor y ejecuta *scripts* cuando se están realizando configuraciones del sistema. Adicionalmente, el *Motor PHP* también proporciona la configuración establecida en el sistema a través de peticiones/respuestas *http/XML*.

Se ha llamado, por tanto, *Motor PHP*, al conjunto de *scripts* escritos en lenguaje *php* que hacen de interfaz, entre los diferentes módulos y elementos, y que permiten realizar las diferentes configuraciones y consultas del sistema, tanto de las *tablas* de la *base de datos* que almacena las configuraciones como del almacenamiento y gestión de archivos multimedia.

El mayor desarrollo del *Motor PHP* se encuentra en el servidor *ESCC* para la configuración y gestión del contenido. El funcionamiento lógico del *Motor PHP* es muy específico a cada *script* determinado; sin embargo, todos estos *scripts* llevan asociado un comportamiento o funcionamiento en común, estableciendo un pequeño estándar en la petición (*métodos GET/POST*) y recepción de datos (formato de datos devueltos en *XML*).

5.4.3 Configuraciones y Aplicaciones en el ESCC para Proveer la Configuración y el Contenido a los ERCs

En este apartado se van a exponer dos aspectos diferenciadores del ESCC. Por una parte, se presentarán las configuraciones más representativas de este equipo para proveer las funcionalidades de la *arquitectura funcional* finalmente desarrollada (*Marco* para establecer la configuración) y, por otra parte, se expondrá el funcionamiento específico de los elementos característicos de éste (*Marco* para proporcionar el funcionamiento).

En la *Interfaz Gráfica de Usuario* se provee una herramienta al usuario administrador para modificar las tablas de la *base de datos* y, por tanto, el comportamiento del sistema en su conjunto. Con la *Interfaz Gráfica de Usuario* se podrá seleccionar qué contenido es emitido por cada ERC o se podrá seleccionar las *referencias NTP*, *dirección IPv4* del servidor o servidores *DNS*; por tanto, con ella se podrán modificar, tanto parámetros del *sistema operativo* sobre el que descansa el sistema, como parámetros concretos de la *Aplicación de Difusión Selectiva*.

Así, los siguientes sub-apartados presentarán los requisitos *software* en el ESCC, las configuraciones necesarias del *sistema operativo* y las configuraciones de ciertas aplicaciones específicas para alcanzar el comportamiento de la *arquitectura funcional* mediante el *Marco* diseñado. Finalmente, se presentará el funcionamiento lógico de los elementos más importantes de este equipo dentro de la *arquitectura funcional*.

5.4.3.1 Sistema Operativo del ESCC

El *sistema operativo* finalmente instalado en el equipo que realiza las tareas de servidor ha sido el *sistema operativo Linux Ubuntu Server 10.10*. Sin embargo, también se llegó a testear, gran parte del funcionamiento del sistema, sobre las versiones de *Ubuntu Server 9.04*, *9.10* y *10.04*.

5.4.3.2 Secuencia de Configuración en el Inicio del ESCC

Se ha programado un *script* que ejecuta las aplicaciones necesarias y aplica las correspondientes configuraciones almacenadas en archivos de sistema al iniciar el ESCC.

Este *script* se ha denominado `inicioServidor.php`. Para ejecutar la secuencia de comandos se configura un enlace simbólico al *script* `inicioServidor.php`, en nivel 2 de ejecución, al iniciar el *sistema operativo Ubuntu Server* en el ESCC. Para ello, con permisos de administrador del sistema, escribimos a través de la consola:

```
pfc@escc:~$ sudo ln -s /opt/lampp/htdocs/div-dat/div-local/INICIO_PHP/inicioServidor.php /etc/rc2.d/S92inicioServidor.php
```

Así, haciendo uso de la aplicación *php* (provista por *XAMPP*) para ejecutar el *script* (`#!/opt/lampp/bin/php`), realizamos las operaciones necesarias para dar de alta las interfaces de red, tal y como se ha comentado que es necesario realizar en el apdo. 5.4.2.2. Así, cada vez que se inicia el sistema le asignamos la configuración lógica, almacenada en la correspondiente tabla de la *base de datos*, en la interfaz física *eth0*.

5.4.3.3 Configuraciones Específicas del ESCC

Antes de realizar las explicaciones de cómo se ha alcanzado el funcionamiento de la *arquitectura funcional* del servidor, se muestran las configuraciones más significativas concretas al ESCC. Estas configuraciones son relativas al *sistema operativo* o a las aplicaciones utilizadas para establecer las funcionalidades lógicas del equipo.

5.4.3.3.1 Configuraciones XAMPP

Dentro de la configuración de *XAMPP*, en concreto de *Apache*, se configura el archivo ***httpd.conf*** (`/opt/lampp/etc/httpd.conf`) para que el servidor web atienda conexiones también en el puerto *TCP 8080*. Se ha programado que las peticiones desde los archivos *swf* se realicen a través de un puerto diferente al que lo hace el navegador por defecto (*TCP 80, 443*). Esto es poco determinante en el sistema, pero se ha decidido así para diferenciar y poder darle un tratamiento especial en futuras implementaciones.

```
listen 8080
```

También se han realizado unos pequeños cambios dentro de *XAMPP* a nivel de la configuración *php*. En concreto se han modificado algunos parámetros del archivo ***php.ini*** (`/opt/lampp/etc/php.ini`), para proveer ciertas características en la configuración del *ESCC*. Concretamente en el equipo se ha de proveer el tamaño de los archivos que se pueden subir al servidor a través de la *Interfaz Gráfica de Administración del Usuario*. Se ha seleccionado que sea un *1 Gigabyte (1 GB)*, aunque puede considerarse que es un valor excesivamente alto, se ha establecido así ante la necesidad de difundir vídeos de alta definición; no obstante, este parámetro es fácilmente modificable:

```
upload_maxfile_size=1024 M;  
post_max_size=1024 M;
```

Este último parámetro tiene una relación con el tiempo que el servidor permite que una conexión esté activa al cargar un fichero desde una página *php*, ya que un *script* estará ejecutándose un tiempo que no debe ser indefinido. En el sistema, cuando se cargan archivos al servidor, se mantendrá la ejecución del *script php* aproximadamente el tiempo que se tarde en enviar el archivo al servidor, el cual dependerá del tipo de conexión que se tenga a éste. El *script php* que adquiere el archivo, como parte del *Motor PHP*, se denomina *carga_archivos_loc.php* (carga los archivos en el servidor) y se llama desde la aplicación *swf*: *Gestion_Archivos.swf*; la cual se incluye como elemento embebido en el *archivo php base* *gestionArchivos.php*.

Por tanto, considerando que se tenga una conexión mínima de *256 kbps*, el tiempo necesario para subir al servidor *1 GB*, son *32768 segundos (9 h 6 m 8 s)*. Se ha establecido el tiempo máximo que el *php* ejecuta un *script* a dicho valor:

```
max_execution_time = 32768;  
max_input_time= 32768;
```

La suposición que se está aplicando en estas configuraciones es que estamos cargando un solo archivo de forma simultánea y que el tiempo en finalizar el *script* que carga los archivos es despreciable en comparación con el tiempo de carga del archivo.

5.4.3.3.2 Configuraciones para la Sincronización de Hora en el ESCC

Se ha comentado en varios apartados (ver apdo. 4.5 Sincronización de la Hora) la necesidad de sincronizar la hora entre los diferentes equipos que forman parte del *Marco* diseñado. Dependiendo del rol de los equipos se adquiere la referencia de hora desde un servidor externo al sistema (*referencias NTP externas*) en el *ESCC* o se sincroniza, en el caso de los *ERCs*, con el servidor principal *ESCC* a través del protocolo *ntp*. En el *ESCC*, al ser también el proveedor de hora para los *ERC*, es necesario ejecutar el servicio servidor *ntp* al iniciar el sistema (*ntpd*), el cual se configura de forma automática al instalar *ntp*. Por tanto, el *ESCC* tendrá un rol de *cliente (ntpd)* y también de *servidor ntp (ntpd)*.


```
pfc@esc:~$ sudo apt-get install ntp ntpdate
```

Las *referencias NTP* externas se configuran a través de la interfaz web del *ESCC* y se mantienen almacenadas en su particular tabla de la *base de datos*. Estas referencias deben ser accesibles, a nivel de conectividad, desde el *ESCC* para establecer la hora en el sistema. Cuando se modifica la configuración relativa a las *referencias NTP Externas* en el *ESCC*, se edita también el archivo (`/etc/ntp.conf`) que provee la configuración y que marcan el comportamiento del protocolo *ntp* :

```
pfc@esc:~$ cat /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
server <ref_NTP_1>
server <ref_NTP_2>
server <ref_NTP_3>
server 127.127.1.1 minpoll 4
fudge 127.127.1.1 stratum 10
```

La edición del archivo se realiza a través del módulo '*HTML y Config. LOCAL*' del *archivo php base* encargado de configurar las *referencias NTP Externas* en la *base de datos*: *NTPServidor.php*. Como se aprecia en la configuración mostrada, no se ha establecido ningún tipo de restricción a la hora de realizar peticiones de sincronización al *ESCC* y se ha dejado como línea de trabajo futuro configurar las restricciones al acceso en la sincronización de la hora (ver apdo. 10.2 Líneas de Trabajo Futuro).

La primera línea indica el archivo donde se guardará el factor de corrección necesario para mantener el reloj sincronizado. El modificador `minpoll 4` indica que los pedidos de sincronización deberán realizarse cada 2^4 segundos, es decir 16 segundos. Este parámetro puede variar entre los valores 4 (16 segundos) y 17 (36,4 horas). Así, se logra reducir significativamente el período de sincronización inicial del protocolo *NTP* para esta referencia [101]. Debido a que esta referencia no es exacta, la siguiente línea (`fudge 127.127.1.1 stratum 10`) indica que se considere a la misma como de *Stratum 10*. Como en este caso (`server 127.127.1.1`) estamos utilizando el reloj interno del equipo en cuestión, que es eminentemente impreciso, es importante que el *Stratum* de esta referencia sea siempre mayor que cualquier otra referencia más precisa [101] [102].

Posteriormente a la edición del archivo se reinicia el *servidor* y *cliente ntp*. De esta forma la configuración con el *servidor ntp* se aplica cuando arranca el sistema o se cambia la configuración de las *referencias NTP Externas* en el caso del *ESCC*. Como aspecto destacable se ha de indicar que si no se ha configurado un servidor (`/etc/resolv.conf`), en el *cliente DNS* de este equipo, la *referencia NTP Externa* debe ser una *dirección IPv4*.

Para finalizar este sub-apartado, tan sólo mencionar que los servidores de referencia en el *ESCC* son los configurados en el propio servidor y si no se ha configurado ninguno, él mismo proporciona la hora local que tenga almacenada el sistema.

5.4.3.3 Configuraciones para la Sincronización del Contenido Multimedia: *Servidor rsync*

Se ha ideado un sistema para que se tengan almacenados los archivos en cada *ERCs* de forma local. Cada *ERC* puede almacenar los archivos localmente para así realizar las peticiones (*HTTP Streaming*) a *localhost* en vez de a la *dirección IPv4* del *ESCC* con el objetivo de ahorrar *ancho de banda*. Esta técnica se ha denominado *arquitectura funcional Auto-Proxy* dentro del contexto del proyecto. Este es un aspecto importante cuando la calidad de los videos a emitir es considerable.

En el *ESCC* se establece la aplicación como servicio para permitir la sincronización del contenido desde los *ERCs*. Para ello se realizan tres pequeñas tareas:

1. Dentro de la configuración de *rsync* se ha de editar (y crear) el archivo `/etc/rsyncd.conf` para así proveer un propio mecanismo de autenticación y control de acceso. En este punto se crea un servidor público de acceso anónimo:

```
pfc@esc:~$ cat /etc/rsyncd.conf
#Parámetros generales
log file=/opt/lampp/htdocs/div-dat/div-log/rsync.log
#módulos
[multimedia]
    path=/opt/lampp/htdocs/div-dat/div-local/archivos
    coment=Servidor de Contenido ESCC: Archivos Multimedia
    list=yes
    read only=yes
```

2. Se edita el siguiente *script* en con el nombre `inicioRsync.php`, para así ejecutar *rsync* como servicio:

```
#!/opt/lampp/bin/php

<?php
    $salida.=system("sudo /usr/bin/rsync --daemon;");
?>
```

3. Se lanza el *script* en el *Nivel de Ejecución 2*, para que al iniciar el sistema se inicie el servicio:

```
pfc@esc:~$ sudo ln -s /opt/lampp/htdocs/div-dat/div-local/INICIO_PHP/inicioRsync.php /etc/rc2.d/S93inicioRsync.php
```

Como se puede observar en la configuración introducida, cualquier equipo podrá establecer el sincronismo con los archivos multimedia almacenados en el *ESCC*, por lo que se le permitiría, a los equipos con conectividad en la red, acceder a los archivos subidos al servidor; los cuales, podrían estar protegidos con derechos de autor. Se deja, por tanto, el sistema sin establecer un determinado control de acceso; sin embargo, se ha considerado el problema y se deja como mejora y línea de trabajo futuro (ver apdo. 10.2 Líneas de Trabajo Futuro).

5.4.3.3.4 Configuraciones del Servidor de Archivos de Políticas de Socket para las Aplicaciones swf al Iniciar el ESCC

Como veremos, al detallar el modo de implementar la *arquitectura funcional* que le corresponde al *ESCC* (apartado 5.4.3.4), será necesario habilitar un *servidor* que provea el *archivo de política de socket* a las *aplicaciones swf*. De este modo se permite que estas aplicaciones puedan conectarse a determinados puertos vía *socket*. En el contexto de la aplicación, los *archivos de políticas de socket* establecerán los permisos para que los *ERCs* puedan conectarse al *servidor socket*.

Para ejecutar este *servidor de archivos de políticas de socket* al inicio del sistema, se utilizará la misma estrategia que para el resto de aplicaciones y *scripts* que se ejecutan al iniciar el equipo; es decir, mediante un enlace simbólico en el *Nivel de Ejecución 2* del *ESCC*:

```
pfc@esc:~$ sudo ln -s /opt/lampp/htdocs/div-dat/div-local/FlashPolicySocket.php /etc/rc2.d/S94FlashPolicySocket.php
```

5.4.3.3.5 Configuración del Servidor Socket al Iniciar el ESCC

Al igual que el resto de aplicaciones y *scripts*, y siguiendo la misma estrategia, lanzamos el *servidor socket* para que se ejecute al inicio del sistema en el *Nivel de Ejecución 2*. A continuación, se muestra esta configuración según el árbol de directorio establecido en el servidor, realizando la configuración a través de consola:

```
pfc@escc:~$ sudo ln -s /opt/lampp/htdocs/div-dat/div-local/socketServidorPHP.php /etc/rc2.d/S95socketServidorPHP.php
```

Se establece la restricción en el *servidor socket* para que el único que pueda escribir en el *socket* sea el propio *ESCC*. No obstante, los detalles del funcionamiento lógico se comentan en apartados sucesivos (5.4.3.4.2 Servidor Socket en el ESCC).

Para ver el número máximo de conexiones que acepta el sistema debemos comprobar el valor *SOMAXCONN* ya que marca el número máximo de conexiones a un *socket*. En *Linux*, este valor se puede comprobar ejecutando la siguiente instrucción:

```
pfc@escc:~$ sudo sysctl -a | grep -I somaxconn
[...]
net.core.somaxconn = 128
```

Se observa que el número máximo de conexiones es 128. Por tanto, se puede indicar que el sistema soportaría como máximo 126 dispositivos *ERCs*. Esto es debido a que la conexión 1 deberá estar reservada para el *ESCC* cuando ejecute órdenes de escritura sobre el *socket* y, al menos, la conexión 128, debería estar disponible para el equipo que accede a la *Interfaz Gráfica de Usuario (EACCC)*. Así, el *EACCC* necesita conectividad al *socket* para monitorizar el sistema de forma remota y se ha de indicar que será este equipo el que indique al *ESCC* que debe enviar ordenes al *socket*; no obstante, no las enviará el directamente en la conexión *socket*, sino que lo hará el propio *ESCC* mediante la petición de ejecución de un *script php* (*actualizar_socket.php*). El *servidor socket* posteriormente, distribuirá la orden al resto de equipos.

Para modificar el número máximo de conexiones *socket* de red que puede aceptar el sistema se puede añadir la siguiente línea al fichero */etc/sysctl.conf*:

```
net.core.somaxconn=<N_CONEX>
```

Donde *<N_CONEX>*, será el número de conexiones máxima que aceptara el sistema por cada *socket*. También se puede variar este parámetro del sistema, directamente, con la siguiente instrucción:

```
pfc@escc:~$ sudo sysctl -w net.core.somaxconn=<N_CONEX>
```

Se ha aceptado que 126 equipos *ERCs* son más que suficientes para el propósito y ámbito del *Marco* implementado. Se ha considerado que podrían ser otros los factores que pudieran reducir o aumentar este valor; atendiendo a, por ejemplo, el *ancho de banda* de la red contratada (para proporcionar acceso a estos equipos) o el número de licencias distribuidas en una situación comercial.

5.4.3.4 Lógica Funcional del ESCC en el Marco de Aplicaciones de Difusión Selectiva

El desarrollo del *ESCC*, al igual que el de los *ERC*, está dividido en capas independientes. Por lo tanto, en este sub-apartado primero vamos a describir la interacción entre las capas encargadas de proporcionar las funcionalidades en el *ESCC*.

La otra interacción importante en el funcionamiento del *ESCC*, que también se va a describir en este sub-apartado, es la comunicación con los *ERCs*, pues la configuración del contenido a mostrar en los *ERCs* se hace desde el *ESCC*. Las modificaciones de la configuración que afectan a los *ERCs* se transmiten mediante el envío de órdenes, a través del *servidor socket*, que definen qué debe hacer cada *ERC* ante las variaciones de la configuración. Por lo tanto, también describiremos la interacción entre el *ESCC* y los *ERCs*.

5.4.3.4.1 Plataforma de Configuración del Marco en el *ESCC*.

Como se ha explicado en el apartado 5.4.2.3.1 Secuencia Lógica de Funcionamiento de la Aplicación Web de Configuración (Interfaz Gráfica de Usuario), se carga, de forma general, un objeto *swf* en las pantallas de configuración del *ESCC*. Estos objetos son aplicaciones escritas en *ActionScript 3* y compiladas en *Adobe Flash Professional CS3*, de modo que se pueden ejecutar en un navegador si se tiene instalada el plugin *Flash Player* que está disponible de forma gratuita para los principales navegadores y *sistemas operativos*.

La interacción y esquema de funcionamiento de las capas, en lo relativo a la configuración del *ESCC*, se muestra en la Ilustración 37.

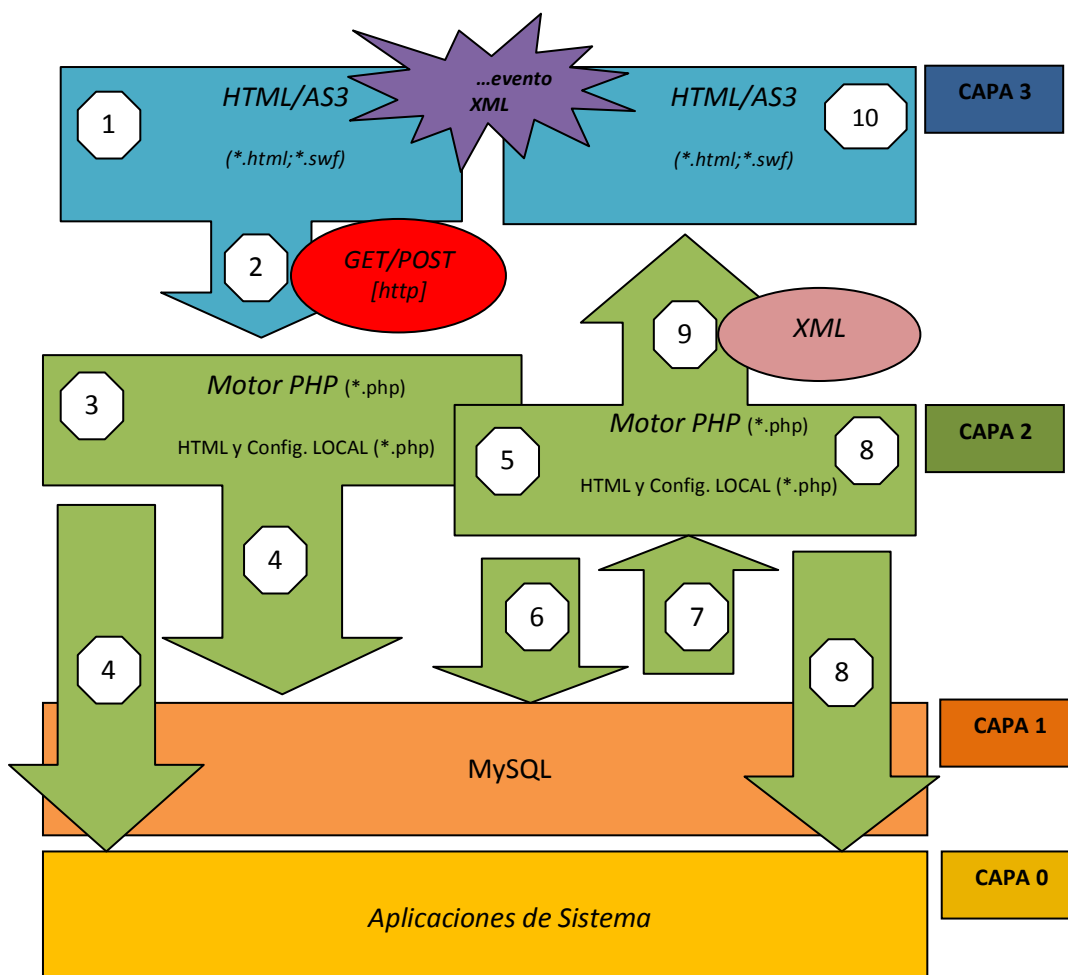


Ilustración 37.- Esquema general del funcionamiento modular en capas, desarrollado en el *ESCC*, de la Arquitectura Software para la configuración de la aplicación de Difusión Selectiva.

La importancia del esquema desarrollado radica en la separación de varias tecnologías diferentes, las cuales son independientes y permiten una interacción entre ellas. Esta separación ha permitido definir diversas capas o niveles, los cuales se comunican entre sí

estableciendo el envío de información. A continuación, se hace una definición algo más precisa de cada uno de las capas que intervienen:

- **CAPA 3:** Es la capa más próxima al usuario y está compuesta por aplicaciones *swf*. Este nivel se comunica con el nivel inferior realizando peticiones *http* a la *Capa 2* y recibiendo a su vez información formateada en *XML*. Esta capa está almacenada en el *ESCC* pero realmente se ejecuta en el navegador del administrador (en el *EACCC*), mientras que el resto de capas se ejecutan en el *ESCC*.
- **CAPA 2:** Es la parte del sistema que se encarga de transformar los datos almacenados en la *base de datos* y que hemos definido como *Motor PHP*. Los datos se devuelven a la *Capa 3* en un formato comprensible y que se ha establecido con anterioridad (datos *XML*). Las peticiones a la *Capa 2* se realizan mediante peticiones *http* al puerto 8080. La *Capa 2* también establece una interacción directa con las *aplicaciones de sistema*: la *Capa 0*.
- **CAPA 1:** Es la *base de datos* del equipo y es donde se almacenan los datos relativos a la configuración de la *aplicación de Difusión Selectiva*. También se almacena de forma persistente, la configuración necesaria del *sistema operativo* para alcanzar, entre otras, conectividad entre los diferentes equipos que forman el sistema. En esta capa descansa el *modelo de datos* de la aplicación.
- **CAPA 0:** Son las *aplicaciones de sistema* en sí. En concreto son archivos de configuración de las aplicaciones que gestionan los recursos del *sistema operativo* y de otras aplicaciones instaladas; fundamentalmente relativas a los recursos relacionados con la conectividad y los protocolos de red utilizados.

De forma general, las aplicaciones *swf* deben interactuar con las *capas* inferiores cuando el usuario ha dado una orden (se produce un evento) desde la *Interfaz Gráfica de Usuario*. Se produce la siguiente secuencia, cuyos pasos más representativos se han etiquetado en la Ilustración 37:

- 1 Se produce un evento *FLASH*: se pulsa algún botón de la *Interfaz Gráfica del Usuario*, en el que se quiere actualizar, por ejemplo, un formulario.
- 2 Mediante el método *GET/POST*, las *aplicaciones swf* envían los parámetros de configuración a un *script php* mediante peticiones *http*. Se ha implementado que las peticiones se realicen mediante el método *GET*; sin embargo, el *Motor PHP* está preparado para aceptar esas mismas peticiones mediante el método *POST*.
- 3 El *Motor PHP* procesa los datos que se han enviado, los cuales los transforma inicialmente en variables.
- 4 Según los datos que se han enviado y a los *script* a los que se les ha realizado la petición, se modifican las tablas de la *base de datos*. Este paso no siempre ocurrirá, ya que podría haber peticiones desde la *aplicación swf* que sólo requieran consultas. Podría ocurrir también que se requiera la ejecución de alguna *aplicación de sistema*.
- 5 Se procesan y analizan los datos recibidos desde el *Motor PHP* antes de realizar la consulta.
- 6 Se realiza la consulta a la *base de datos* a través del *Motor PHP*.
- 7 Se reciben los datos de la *base de datos* a través del *Motor PHP*.
- 8 Se procesan los datos obtenidos de la *base de datos*. Se ejecutan aplicaciones, *scripts* o se modifican archivos del sistema dependiendo de la necesidad o de la petición u orden recibida.
- 9 Se forma una respuesta en formato *XML*.
- 10 La *aplicación swf* (*archivo Flash*) recibe la información en formato *XML*, de modo que se produce un evento. En la presentación al usuario se modifica sólo la información procesada y se sigue con la ejecución de la aplicación *swf* (*Flash*). De esta forma, se

establece una comunicación *asíncrona*, ya que no se carga de nuevo la página, sino que se actualiza el contenido con los datos recibidos.

La siguiente tabla muestra, a modo orientativo, la relación entre las capas superiores para los diferentes *scripts* y aplicaciones en el *ESCC*:

Archivo PHP Base	Capa 3	Capa 2	Capa 1/Capa 0
	SWF	Motor PHP/Config. PHP	Modelo Datos
Ubic.php	Ubic.swf	edita_ubic.php	_ubicaciones [BBDD] _config_ubic [BBDD] _clientes [BBDD] orden_socket() [PHP]
zonasEsc.php	1_Esc_Zonas.swf	config_esc.php	_esc_conf [BBDD] _esc_reproducido [BBDD]
		muestra_fondos.php	m_multimedia_local [BBDD]
listasZonas.php	2_Zonas_Listas.swf	config_esc.php	_esc_conf [BBDD] _esc_reproducido [BBDD]
		programa_listas.php	_programacion_zonas [BBDD]
		muestra_listas.php	_programacion_zonas [BBDD] l_listas [BBDD]
		edita_listas.php	l_listas [BBDD]
archivosListas.php	3_Listas_Archivos.swf	muestra_listas.php	_programacion_zonas [BBDD] l_listas [BBDD]
		edita_listas.php	l_listas [BBDD]
		edita_archivos.php	m_multimedia_local [mostrar archivos] [BBDD]
EquiposUbic.php	Equipos_Ubic.swf	edita_equipos.php	_ubicaciones [BBDD] _clientes [BBDD] orden_socket() [PHP]
gestionArchivos.php	Gestion_Archivos.swf	Carga_archivos-loc.php	m_multimedia_local [BBDD] orden_socket(sincronizar) [APLIC] archivos/VIDEO/ [FICH] archivos/IMAG/ [FICH] archivos/MEDIA/ [FICH] files [APLIC]
		Borra_archivos-loc-B.php	m_multimedia_local [BBDD] _ubicaciones [BBDD] l_listas [BBDD] archivos/VIDEO/ [FICH] archivos/IMAG/ [FICH] archivos/MEDIA/ [FICH] orden_socket(actualizar ubi) [PHP] unlink [APLIC] files [APLIC]
buscaArchivos.php	Buscar_Archivos.swf	edita_ubic.php	_ubicaciones [mostrar] [BBDD]
		edita_archivos.php	m_multimedia_local [mostrar archivos] [BBDD]
		buscar_archivos.php	_ubicacion_listas _lista_archivos m_multimedia_local [BBDD]
		buscar_todas_listas.php	_ubicaciones [BBDD] _ubicacion_listas
actualizaCont.php	Actualizar_Equipos.swf	edita_ubic.php	_ubicaciones [mostrar] [BBDD]
		actualizar_socket.php	orden_socket() [PHP]
actualizaPersonalizado.php	Actualizar_Personalizado.swf	edita_equipos.php	_ubicaciones [BBDD] _clientes [BBDD] orden_socket() [PHP]
IPServidor.php	N/A	IPServidor.php StatusCambioIP.php	_config_IP [BBDD] _config_DNS [BBDD] /etc/network/interfaces [FICH][APLIC] /etc/resolv.conf [FICH][APLIC] /etc/init.d/networking [APLIC] Route [APLIC] Ifconfig [APLIC] Ifdown [APLIC] Ifup [APLIC] Ping [APLIC]
DNSServidor.php	N/A	DNSServidor.php	_config_DNS [BBDD]

Archivo PHP Base	Capa 3	Capa 2	Capa 1/Capa 0
	SWF	Motor PHP/Config. PHP	Modelo Datos
		StatusDNS.php	_config_IP [BBDD] /etc/resolv.conf [FICH] [APLIC] ping [APLIC]
NTPServidor.php	N/A	NTPServidor.php	_config_NTP [BBDD] /etc/ntp.conf [FICH][APLIC] /etc/init.d/ntp [APLIC]
cambiaContra.php	Cambia_Contra.swf	cambia_contra.php	_usuarios [BBDD]
StatusCambioIP.php	N/A	StatusCambioIP.php	_config_IP [BBDD] Ifconfig [APLIC] Route [APLIC] Ping [APLIC]
StatusDNS.php	N/A	StatusDNS.php	_config_DNS [BBDD] /etc/resolv.conf [FICH] [APLIC] ping [APLIC]
StatusNTP.php	N/A	StatusNTP.php	_config_NTP [BBDD] Host [APLIC] Ping [APLIC]
reiniciarServidor.php	N/A	reiniciarServidor.php	Shutdown [APLIC]
Descarga_Software.php	Descargas_Windows.swf Descargas_Linux.swf	N/A	N/A
ListadoArchivos.php	N/A	ListadoArchivos.php	archivos/VIDEO/ [FICH] archivos/IMAG/ [FICH] archivos/MEDIA/ [FICH]
mediaWiki.php	N/A	mediawiki/index.php	MEDIAWIKI [APLIC] [BBDD]

Tabla 14.-. Resumen de Interacción entre capas de los archivos y aplicaciones desarrolladas en el proyecto para la configuración del ESCC.

En la tabla anterior se muestra la interacción entre las diferentes aplicaciones y *scripts* dentro del *Marco* desarrollado, además de hacer referencia a algunas aplicaciones y archivos de directorios a los cuales se accede.

5.4.3.4.1.1 Esquema General de ActionScript: Aplicaciones SWF en el ESCC

La *Interfaz Gráfica de Administración del Usuario* basa gran parte de su funcionamiento en las *aplicaciones swf* desarrolladas. *Adobe Flash* fue diseñado expresamente para proveer interacción con el usuario final y ha sido la tecnología escogida para desarrollar la *interfaz gráfica de usuario* en el *ESCC*.

Dependiendo del *archivo php base* que se está presentando, se cargará una u otra *aplicación swf* (ver Tabla 14), a la cual se le pasarán algunos parámetros (*Ubicación, IPv4 del ESCC, puerto socket, puerto petición web*) y que son utilizados para realizar las peticiones que se realizan al *Motor PHP* (desde las aplicaciones ejecutadas mediante *Flash Player*) y realizar la conexión al *servidor socket*.

A continuación se presenta un esquema muy genérico del funcionamiento de estas aplicaciones y nos sirve para comprender en detalle la estrategia de utilizada:

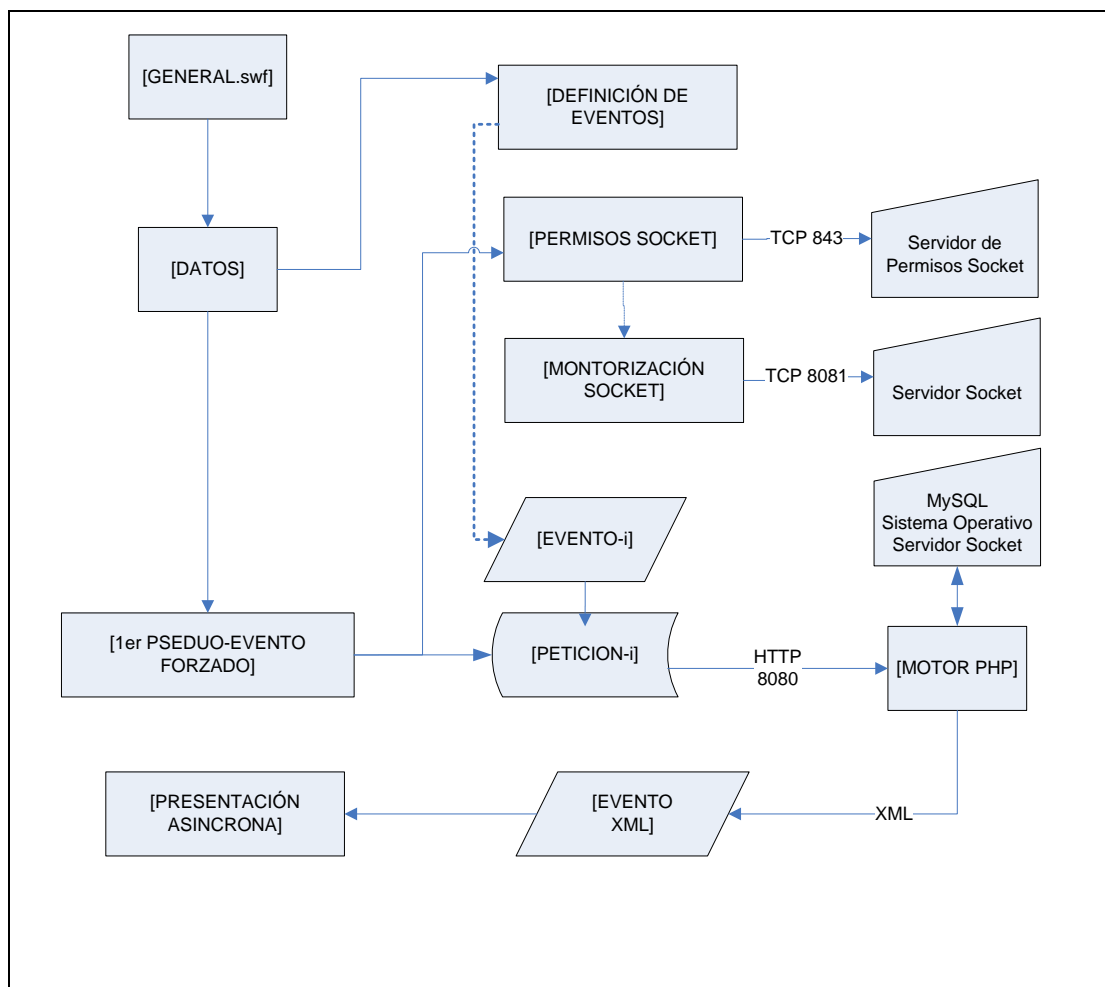


Ilustración 38.-. Esquema Genérico de Funcionamiento de las Aplicaciones SWF en el ESCC.

El esquema de funcionamiento es reutilizado en la mayoría de las aplicaciones implementadas; sin embargo, se ha de notar que el número de eventos registrados son variables y distintos en cada aplicación. También son varios, en algunas ocasiones, los scripts del *Motor PHP* llamados, cada vez que se produce un determinado *evento-i* en la *Interfaz Gráfica de Usuario*, en concreto en la *aplicación swf*.

Los puertos utilizados por estas *aplicaciones swf* son básicamente 3 y se utilizan para establecer las comunicaciones entre los equipos:

- Peticiones de archivo de políticas de socket. En este caso se utiliza el puerto *TCP 843*. Se obtiene el fichero de permisos desde donde las *aplicaciones swf* pueden identificar si tienen permisos para acceder a un determinado *socket* en un determinado equipo. Como se explica de forma detallada en el apdo. 5.4.3.4.2 Servidor de Archivo de Política Socket para Flash Player, esta conectividad es necesaria a partir de la versión 9.0.124.0 de *Flash Player*.
- Petición de conexión al servidor socket. El puerto utilizado es el *TCP 8081*. Las *aplicaciones swf* de la *Interfaz Gráfica del Usuario* y que se ejecutan en el navegador del *EACCC*, se conectan al *servidor socket* en el *ESCC* para realizar una monitorización de las órdenes allí establecidas. El funcionamiento detallado del *servidor socket* se presenta en el apdo. 5.4.3.4.2 Servidor Socket en el *ESCC*.

- *Peticiones http hacia el Motor PHP*: Se utiliza el puerto *TCP 8080* para diferenciar estas peticiones en el futuro, de modo que pueda realizarse un tratamiento diferente a nivel del servidor.

5.4.3.4.1.2 Motor PHP en el ESCC

El *Motor PHP* tiene como función principal consolidar, consultar y actualizar la configuración de forma persistente en las tablas de la *base de datos* y en la configuración de las aplicaciones del sistema.

El *Motor PHP* devuelve la información formateada en *XML* a través del protocolo *http*, para que así puedan ser interpretadas por la aplicación *swf* (que se ejecutan en el *EACCC*). Las peticiones, también en *http* y utilizando el método *GET*, se realizan desde las aplicaciones *swf* que están escritas en *ActionScript 3 (AS3)*, almacenadas en el *ESCC* y ejecutadas en el *EACCC*. Así, cuando se realiza la configuración del sistema se realizan llamadas mediante el navegador web (*https*) que devuelve código *html* con aplicaciones *swf* embebidas. Estas aplicaciones *swf*, se encuentran en la *Capa 3* y son las que interactúan con la *Capa 2*, en donde se ha programado el *Motor PHP*. Ante las peticiones de la *Capa 3*, el *Motor PHP*, realiza peticiones a la *Capa 1* de modo que cada *script* del *Motor PHP* interpreta las respuestas (*MySQL*) y las formatea en formato *XML* para que la *Capa 3* pueda interpretarlas.

A todos los *script* del *Motor PHP* se les pasa, mediante el método *GET*, la ‘acción’ a realizar y un conjunto de parámetros definidos para esa acción. A continuación, se muestra una tabla resumen de los *scripts* del *Motor PHP* desarrollados con un pequeño resumen de las acciones que realizan y la información devuelta a la aplicación que realiza la petición:

Motor PHP	Acciones: Descripción
edita_ubic.php	<p>[Insertar](Ubicación): Inserta una nueva <i>Ubicación</i>. [Cambiar](Ubicación): Cambia el nombre de una <i>Ubicación</i>. [Eliminar](Ubicación): Elimina una <i>Ubicación</i> del sistema. Configura todos los equipos de la <i>Ubicación</i> eliminada en la <i>Ubicación</i> ‘defecto’. [Mostrar](Ubicación): Devuelve un archivo <i>XML</i> con las ubicaciones configuradas en el sistema.</p> <p>En todos los casos el <i>Motor PHP</i> siempre devuelve una información formateada en <i>XML</i> con el listado de las <i>ubicaciones</i> configuradas y el número de equipos clientes asignados a cada <i>Ubicación</i>. El <i>Motor PHP</i> establece las órdenes apropiadas en el <i>servidor socket</i> que afectan a las <i>ubicaciones</i> que han sufrido modificación.</p>
edita_equipos.php	<p>[Insertar](ERC): Inserta la <i>dirección IPv4</i> de un nuevo <i>ERC</i> asociándolo a una <i>Ubicación</i>. [Cambiar](ERC): Cambia la asociación de una <i>dirección IPv4</i> de un <i>ERC</i>, a otra <i>Ubicación</i> diferente. Esta opción permite cambiar, por tanto, a un <i>ERC</i> de <i>Ubicación</i>. [Eliminar](ERC): Con esta instrucción se elimina la asociación de la <i>dirección IPv4</i> del <i>ERC</i> con la <i>Ubicación</i> seleccionada. [Editar](ERC): Permite modificar la <i>dirección IPv4</i> asociada a una <i>Ubicación</i> y establecer un nuevo valor. [Mostrar](ERC): Muestra un listado de las asociaciones entre las <i>direcciones IPv4</i> de los <i>ERCs</i> y las <i>ubicaciones</i> dadas de alta en el sistema.</p> <p>En todos los casos se devuelve información formateada en <i>XML</i> con un listado de las <i>ubicaciones</i> y los <i>ERCs</i> asociados a cada <i>Ubicación</i>. En todos los casos también se escribe una orden en el <i>socket</i> para que el equipo actualice la <i>Ubicación</i>. En el caso de editar se escriben dos órdenes, una para la <i>dirección anterior</i> (pase a la <i>Ubicación</i> ‘defecto’) y otra para la nueva <i>dirección IPv4</i>.</p>
config_esc.php	<p>[Editar](Esc): Se actualizan los valores de las Zonas, así como la imagen de fondo del Escenario de una determinada <i>Ubicación</i>. [Cambiar](Esc): Se establece cual de los Escenarios disponibles es el que se establecerá para la visualización de una determinada <i>Ubicación</i>.</p> <p>En todos los casos el <i>Motor PHP</i> muestra la posición y el tamaño de las Zonas configuradas de todos los Escenarios de una determinada <i>Ubicación</i>.</p>
muestra_fondos.php	<p>El <i>Motor PHP</i> devuelve un listado todos los fondos disponibles. Los fondos disponibles corresponderán con todas las imágenes almacenadas en el <i>ESCC</i>.</p>
programa_listas.php	<p>[Insertar](lista): Inserta una nueva <i>lista de programación</i> dentro de la <i>Zona</i> de un determinado <i>Escenario</i> de una <i>Ubicación</i>. Se debe introducir una fecha y que esta no coincida con una lista ya</p>

Motor PHP	Acciones: Descripción
	<p>programada.</p> <p>[Eliminar] (lista): Elimina una lista de reproducción programada de una Zona de un Escenario de una Ubicación.</p> <p>[Editar] (lista): Modifica la programación con nuevos valores de listas. Permite modificar los parámetros (fechas) de una sola lista en cada petición.</p> <p>El Motor PHP devuelve en todos los casos un listado en XML de las listas de reproducción con las fechas de inicio y de fin de la Zona del Escenario de la Ubicación de la que se hace la petición.</p>
muestra_listas.php	<p>[Nueva] (lista): Permite crear una nueva lista para una Ubicación.</p> <p>[Borrar] (lista): Permite eliminar una lista de una determinada Ubicación.</p> <p>El Motor PHP devuelve en formato XML, mediante este script, las listas de reproducción que existen para la Ubicación que se está configurando.</p>
edita_listas.php	<p>[Insertar] (archivo/lista): Esta acción permite insertar un nuevo archivo en la lista de reproducción indicada.</p> <p>[Eliminar] (archivo/lista): Permite eliminar un archivo de una determinada lista de reproducción.</p> <p>[Editar] (archivo/lista): Permite modificar los parámetros de algún archivo de la lista de reproducción.</p> <p>[Mostrar] (archivo/lista): No realiza ninguna acción adicional, simplemente muestra los archivos de una determinada lista de reproducción.</p> <p>El Motor PHP en todos los casos devuelve, en formato XML, un listado de los archivos de reproducción de la lista junto con los parámetros específicos del archivo en esa lista. Siempre hay que pasarle como parámetro el nombre de la lista de reproducción de la Ubicación. Si se quiere editar, insertar o eliminar se debe pasar al menos el id del archivo en cuestión.</p>
buscar_todas_listas.php	<p>Este script muestra un listado de todas las listas disponibles independientemente de la Ubicación a la que corresponda. No necesita parámetros de entrada. Al igual que el resto de scripts del Motor PHP, devuelve la información almacenada en el modelo de datos mediante un esquema XML.</p>
edita_archivos.php	<p>[Insertar] (archivos): No se implementa aquí. Ver carga_archivos-loc.php</p> <p>[Eliminar] (archivos): No se implementa aquí. Ver borra_archivos-loc-B.php</p> <p>[Mostrar] (archivos): No realiza ninguna acción adicional. Muestra los archivos disponibles.</p> <p>El Motor PHP devuelve en todos los casos y en formato XML, un listado con todos los archivos subidos al servidor mediante la aplicación swf. La tabla m_multimedia_local contiene todos los archivos multimedia que están disponibles en el sistema y que se almacenan de forma local en el disco mediante la subida de archivos al ESCC.</p>
carga_archivos-loc.php	<p>Permite la carga de archivos en el servidor y registra el archivo en la tabla m_multimedia_local. Indica la orden de sincronización escribiendo en el servidor socket para que los ERCs sincronicen el contenido de forma local con el ESCC.</p>
borra_archivos-loc-B.php	<p>El script permite recibir mediante el método GET/POST el identificador del archivo multimedia que se desea eliminar del sistema. El Motor PHP establece un bucle para explorar todas las listas de todas las ubicaciones e indica orden de actualizar la Ubicación afectada en el servidor socket (si fuera necesario). Adicionalmente, envía la orden de sincronización de contenido a todos los ERCs para que estos eliminen el archivo almacenado de forma local.</p>
buscar_archivos.php	<p>El Motor PHP, mediante este script, permite realizar una búsqueda de los archivos configurados en el sistema atendiendo a diversos parámetros para filtrar la búsqueda. Estos filtros pueden ajustarse a través de la Ubicación, el nombre de la lista en la que se encuentra el archivo o el nombre del propio archivo. El script devuelve en formato XML un listado de los archivos, con información de la Ubicación y la lista en la que está configurado. La presentación de esta información resulta útil al administrador del sistema.</p>

Tabla 15.-. Resumen de scripts php que forman el Motor PHP.

Los scripts del Motor PHP de la tabla anterior devuelven, formateada en XML, la información que se ha resumido en la descripción de la Tabla 15. A continuación, se muestra un ejemplo de una petición establecida y del resultado obtenido:

Ejemplos de peticiones Http

El script programa_listas.php, devuelve un listado de las listas de reproducción programadas para una Zona de un Escenario de una Ubicación concreta. Los parámetros se envían mediante el método GET, aunque el script también está programado para recibirlo por el método POST:

```
[mostrar]
programa_listas.php?accion=mostrarr&ubicacion=defecto&esc=3&zona=B

[editar]
```

```

programa_listas.php?accion=editar&id_programacion=158&esc=3&zona=B&nombre=defecto
&inicio=2008-02-28&fin=2008-03-31&algoritmo=1&transicion=1

[insertar]
programa_listas.php?accion=insertar&ubicacion=defecto&id_programacion=0&esc=3&zona=B&nombre=defecto2&inicio=2008-02-05&fin=2008-03-31&algoritmo=2&transicion=2

[eliminar]
programa_listas.php?accion=eliminar&id_programacion=143&esc=3&zona=B

```

Ejemplo de respuesta en formato XML

Un ejemplo, de la respuesta devuelta por el ESCC es la siguiente:

```

<datos_programacion tabla=_programacion_zonas esc=3 zona=B>

  <valores select=on id_programacion=163 esc=3 zona=B nombre=defecto2 incio=2010-02-05 fin=2010-03-31 algoritmo=2 transicion=2>
  <valores id_programacion=162 esc=3 zona=B nombre=defecto2 incio=2010-02-05 fin=2010-03-31 algoritmo=2 transicion=2>
  <valores id_programacion=158 esc=3 zona=B nombre=defecto2 incio=2010-02-28 fin=2010-03-31 algoritmo=1 transicion=1>

</datos_programacion>

```

5.4.3.4.2 Servidor Socket en el ESCC

Como se ha comentado, la segunda interacción importante que define el funcionamiento del ESCC es su comunicación con los diferentes ERCs a través de un **servidor socket**.

La función de este *servidor socket* es la de informar a los ERCs que han sufrido un cambio en su configuración y, así, actualicen la presentación de contenido enviando nuevas peticiones de la nueva configuración al ESCC. Por ejemplo, la comunicación del ESCC con los ERC se produce cuando se realizan cambios de configuración en las *ubicaciones*. Cuando se quita (sin eliminarlo del sistema) un archivo de una *lista de reproducción* programada en una *Ubicación*, se ha de avisar a todos los equipos asociados para que adquieran la nueva configuración. En esta situación no es demasiado crítico ya que el archivo está presente en el sistema. Aún así, hay otras situaciones en las que se convierte en crítica. Por ejemplo, si se elimina finalmente un archivo del servidor y éste está programado en una *lista de reproducción* de la *Ubicación* de un ERC, éste no podrá acceder al archivo y la pantalla quedará en negro sin mostrar el archivo programado porque el archivo ha sido eliminado del sistema. Para solucionar este problema y otros, se ha desarrollado una comunicación entre los equipos basada en *sockets*. El ESCC escribe vía *socket* en el equipo cada vez que se produce una modificación relevante (o se indica explícitamente a través de una orden manual desde la interfaz gráfica), de modo que todos los ERCs reciben la información de que se ha modificado la configuración. Los ERCs deben discriminar si se vieran afectados por la orden recibida. Así, siguiendo con el ejemplo anterior, cuando se elimina un archivo del servidor, se elimina de la configuración de todas las *listas de reproducción* de todas las *ubicaciones* y se indica, a los diferentes ERC que son miembros de las *ubicaciones* implicadas, que deben actualizar el contenido realizando una nueva petición de configuración al ESCC. Si la *Ubicación* no tenía el archivo programado en ninguna de sus *listas* no se indica que esa *Ubicación* tenga que actualizarse. Por esta y otras situaciones es necesario establecer un mecanismo de comunicación.

Se han definido varias órdenes en el servidor que los ERCs pueden interpretar para solucionar la problemática comentada en el párrafo anterior:

- La orden más importante que se escribe en el *servidor socket* es la de actualización de una determinada *Ubicación*. De este modo, cuando se modifica la

configuración de una *Ubicación* los equipos que pertenecen a ésta solicitan al ESCC la nueva configuración.

- Se ha programado también la orden a nivel de *dirección IPv4*, en vez de *Ubicación*, como orden más concreta a un solo equipo. De este modo, el equipo al reconocer que la orden va dirigida a él mismo solicita al ESCC la nueva configuración.
- Otra de las órdenes que se transmiten desde el ESCC, utilizando este *servidor socket* para enviar la información a todos los ERCs, es la orden de *sincronizar* el contenido multimedia local del ERC, con el contenido multimedia almacenado en el ESCC. Así, se puede agilizar la descarga de contenido de la arquitectura *Auto-Proxy* y se evita espera a las tareas programadas cron (ver apdo. 5.4.4.3.4 Configuración del Cliente rsync).

Una vez se escribe una orden determinada en el *socket* esta información se distribuye a cada ERC mediante el *servidor socket*, el cual gestiona cada conexión con los diferentes ERCs. Se ha programado un *servidor socket* en el ESCC de modo que es éste el que permite la comunicación con el resto de equipos del sistema. El funcionamiento lógico es bastante conceptual y consiste en la propagación del mensaje escrito en el *socket* hacia los ERCs. Así, el funcionamiento es parecido al de un ‘chat’, en el que todos los ERCs están escuchando las órdenes del ESCC; no obstante, y siguiendo con la analogía, sólo el ESCC es el que tiene la capacidad ‘escribir’ y, por tanto, enviar órdenes desde el *socket*. De este modo no cualquier equipo conectado al *servidor socket* puede enviarlas, añadiendo cierto grado de seguridad.

Por otra parte, los ERCs, a partir de una *aplicación swf* que realiza la conexión con el *servidor socket*, interpretan las órdenes (*CONTROL SWF*) y actúan en consecuencia: pidiendo de nuevo la configuración al servidor y/o sincronizando el contenido multimedia.

El *servidor socket* se ha programado en lenguaje *php* y el *script* se lanza al inicio del sistema en el nivel de ejecución **rc2.d** del *sistema operativo Ubuntu Server (10.10)* como se ha comentado en el apdo. 5.4.3.3.5 Configuración del Servidor Socket al Iniciar el ESCC. No obstante, para proporcionar el acceso al *servidor sockets* desde aplicaciones *swf* se deben tener presentes algunas consideraciones de la tecnología *Flash Player*.

5.4.3.4.2.1 Archivos Maestros de Políticas. Acceso a Socket desde Aplicaciones swf.

Dentro de la tecnología utilizada se ha optado, como se ha comentado, por utilizar *ActionScript 3*. El navegador web en cada página de configuración del ESCC carga una *aplicación swf*, programada en *ActionScript 3* y que se ejecutan mediante el *plugin Flash Player*, que presenta una interfaz amigable al usuario. Así, algunas de las aplicaciones *swf* desarrolladas y cargadas en el navegador necesitan poder acceder al *socket* para identificar las órdenes enviadas a los ERCs y de este modo monitorizar que éstas se han enviado de forma correcta. Estas órdenes también se deben interpretar en el propio ERC mediante la *aplicación swf* de control desarrollada: *CONTROL SWF* (ver apdo. 5.4.4.4.3.2 Aplicación Control swf y Clientes Socket en los ERCs)

En los inicios de la popularización de *Flash Player*, se introdujo en la versión 6 los *archivos de política url*, aunque todavía no se admitían los *archivos de política socket*. Las conexiones a servidores de *socket* se autorizaban a través de un *archivo de política* de forma predeterminada en un servidor *HTTP* del puerto TCP 80 del mismo host que el *servidor socket*. *Flash Player 9.0.115* e inferiores todavía admiten esta capacidad, pero en versiones a 9.0.124 y posteriores, como la versión de 10 de *Flash Player*, no admiten este sistema [103] y sólo los *archivos de política de socket* pueden autorizar las conexiones *socket*. Así, a partir de la versión de *Adobe Flash Player 9.0.124.0* no se permite conexiones *socket* directamente a un servidor sin obtener antes, y explícitamente, los *permisos socket* desde el propio servidor. Estos

permisos, están definidos en el **archivo de políticas de socket**. Este aspecto requiere en el *Macro* abrir puertos y ejecutar nuevos servicios con el objetivo de proporcionar y garantizar el acceso a los *socket* desde las *aplicaciones swf*.

Las aplicaciones *swf* que se ejecutan en *Flash Player* buscan un **archivo de política url** denominado *crossdomain.xml* en el directorio raíz del servidor y un **archivo de política de socket** en el puerto *TCP 843*. Los archivos que se encuentran en cualquiera de estos dos lugares se denominan **archivos maestros de política**. Además de especificar los permisos de acceso, el *archivo maestro de política* puede contener también una sentencia *meta-policy*. Una *meta-política* (*meta-policy*), en un *archivo de política url*, especifica las *urls* que pueden contener archivos de política. La *meta-política* predeterminada de los *archivos de política URL* es '*master-only*'; es decir, el único archivo de política permitido en el servidor será */crossdomain.xml*. La *meta-política* predeterminada de los *archivos de política de socket* es '*all*', es decir, cualquier *socket* del host puede servir un *archivo de política de socket* aunque el archivo maestro siempre se encuentra en el puerto *TCP 843* [104].

La versión de *Adobe Flash Player 9.0.124.0*, incluye, por tanto, dos grandes cambios en el tratamiento de los *socket* [105][106]:

1. El primer cambio significativo es que *Flash Player* requiere, a partir de esta versión, *archivos de política de socket* para establecer cualquier conexión *socket*. En versiones anteriores de *Flash Player* era posible conectar a puertos superiores de 1024 si la aplicación *swf* se conectaba al mismo servidor en el cual estaba hospedada. Esta situación coincide con la del presente trabajo, de modo que en versiones anteriores de *Flash Player* (9.0.115.0 e inferiores) no es necesario implementar ninguna descarga del *archivo de políticas de socket*.
2. El segundo cambio es la introducción de una localización centralizada para los *archivos de políticas de socket* que pueden almacenar un archivo de políticas de *socket* maestro.

Adobe ha acordado [103] con la *IANA* (*Internet Assigned Numbers Authority*) [107], aunque a día de hoy no está establecido de forma oficial [108], reservar el puerto *TCP 843* para propósitos de servir *archivos de política de socket* y para introducir una localización centralizada para los *archivos de política de socket*. Así, *Flash Player* habilita un modo de administrar el sistema para definir qué puertos están disponibles a través de la *política maestra* (*principal*) que sobrescribe o anula cualquier otro archivo de política en el equipo.

De forma predeterminada, el acceso a conexiones de *socket* está desactivado, incluso si el *socket* al que se conecta se encuentra en el dominio al que pertenece el archivo *swf*. El acceso a nivel de *socket* es posible si se sirve un *archivo de política de socket* desde cualquiera de las ubicaciones siguientes:

1. Puerto *TCP 843* (la ubicación física del *archivo maestro de política*), de forma predeterminada.
2. El mismo puerto que la conexión de *socket* principal, de forma predeterminada y con menor prevalencia.
3. Un puerto diferente a la conexión de *socket* principal especificado en la propia aplicación. Para servir un archivo de política de *socket* desde un puerto diferente, la aplicación *swf* debe llamar a `Security.loadPolicyFile()`.

A modo de resumen, si *Flash Player 9.0.124.0* (y posteriores versiones) no puede recuperar el *archivo de política maestra socket* desde el puerto 843, entonces realiza las peticiones en el

mismo puerto donde está intentando establecer la conexión. Sin embargo, si un *archivo de políticas* está disponible desde el servicio en el *puerto TCP 843*, entonces *Flash Player* considera a esos permisos como prevalecientes para ese sistema. Es decir, si una aplicación quisiera establecer un *socket* con el *puerto 8081* del servidor, inicialmente realizará una petición del *archivo de políticas de socket* en el *puerto TCP 843*:

- Si no se tuviera éxito, intentaría entonces, de forma automática, recuperar el archivo de políticas de *socket* directamente en el *puerto 8081* e intentaría la conexión con ese *socket* si el *archivo de políticas de socket* contuviese una política permisiva.
- Si se tuviera éxito al intentar descargar el archivo de políticas de *socket* en el *puerto TCP 843*, este archivo prevalecería sobre el descargado en el *puerto 8081* y es uno de los motivos por el que se decide establecer una política de *socket* centralizada en el *puerto TCP 843*.

Cuando un *archivo de política de socket* procede de un número de puerto inferior a 1024, dicho archivo puede conceder acceso a cualquier puerto; pero, cuando un archivo de política procede del puerto 1024 o superior, sólo puede conceder acceso a otros puertos 1024 y superiores. Esta característica fue otro motivo por el que se decidió implementar un *servidor archivos de políticas de socket* sobre el *puerto 843* en vez sobre el propio 8081 (u otro diferente). Es decir, en el futuro no sabemos si necesitaremos puertos inferiores a 1024 para establecer conexiones *socket*, de forma que el *Marco para Aplicaciones de Difusión Selectiva de Contenido Multimedia* se deja lo más abierto posible.

La necesidad de descargar las políticas del servidor, plantea nuevas necesidades en el sistema, ya que la *arquitectura funcional* se basa en que las *aplicaciones swf* se puedan conectar al *servidor socket* en el *puerto 8081*. La solución final adoptada se explica en detalle en el siguiente apartado (5.4.3.4.2.2 Servidor de Archivo de Política Socket para Flash Player) y consiste en la creación de un nuevo *servidor socket* que atiende las peticiones en el *puerto 843* proveyendo el ***archivo de política de socket maestro***, frente a la opción de proveer el *archivo de políticas de socket* en el mismo puerto que el *servidor socket* u otro diferente.

5.4.3.4.2.2 Servidor de Archivo de Política Socket para Flash Player

Como se ha comentado en el subapartado anterior, a partir de la *versión 9.0.124.0* de *Flash Player*, se ha modificado la política de seguridad y acceso a la hora de conectar con los distintos *socket* que pudiera necesitar una aplicación *swf*. Estos cambios en la política de seguridad llevan asociadas consecuencias importantes, hasta el punto de que versiones más modernas de *Adobe Flash Player* instaladas en el navegador pueden provocar un mal funcionamiento de las aplicaciones desarrolladas y almacenadas en un servidor. Esto es debido a que no puedan conectarse a servidores *sockets*, debido a que las aplicaciones no hubieran recibido un *archivo de política* de forma apropiada por no haber sido previsto en versiones anteriores.

La política escogida en la aplicación es, por tanto, que la *aplicación swf* inicialmente busque un archivo de *política socket* en el *puerto TCP 843* del servidor que provee el archivo *swf*; es decir, desde el *ESCC*. Los *archivos de política de socket* presentan la misma sintaxis que los *archivos de política url*, salvo por el hecho de que también deben especificar los puertos a los que concede acceso. Al igual que al *archivo de política url* almacenado en el directorio principal del servidor *http* (ver ANEXO E Archivo de Políticas URL para las Aplicaciones SWF), denominado *archivo maestro de política url (crossdomain.xml)*, al archivo proporcionado en el *puerto 843* se le denomina ***archivo maestro de política de socket***.

Los puertos permitidos se especifican en el atributo `to-ports` de la etiqueta `<allow-access-from>`. Se aceptan como valor los números de puerto únicos, los intervalos de puertos y los comodines (*).

A continuación, se muestra un ejemplo de un archivo de política de *socket*:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<!-- Policy file for xmlsocket://socks.mysite.com -->
<cross-domain-policy>
  <allow-access-from domain="*" to-ports="8081" />
  <allow-access-from domain="*.example.com" to-ports="507,516" />
  <allow-access-from domain="*.example.org" to-ports="516-523" />
  <allow-access-from domain="adobe.com" to-ports="507,516-523" />
  <allow-access-from domain="192.168.101.166" to-ports="*" />
</cross-domain-policy>
```

Para recuperar un *archivo de política de socket* del puerto TCP 843 o del mismo puerto que utiliza la conexión de *socket* principal, no es necesario realizar ninguna llamada adicional; simplemente cuando se realiza la llamada con el método `Socket.connect()` o `XMLSocket.connect()`, *Flash Player* se encarga de realizar la petición del *archivo de políticas de socket* automáticamente: en primer lugar, se comprueba la presencia del **archivo maestro de política** en el puerto TCP 843:

- Si encuentra alguno, comprobará si éste contiene una sentencia de *metapolítica* que prohíba la conexión al *socket* en el puerto de destino. Si el acceso no está prohibido, *Flash Player* buscará en primer lugar la sentencia `allow-access-from` en el *archivo maestro de política*.
- Si no encuentra ninguno, entonces buscará un *archivo de política de socket* en el mismo puerto que utiliza la conexión de *socket* principal, en nuestro caso será el puerto 8081.

Para recuperar un archivo de política de *socket* desde un puerto diferente habría que llamar al método `Security.loadPolicyFile()` con la sintaxis especial "*xmlsocket*", como en el ejemplo siguiente:

```
Security.loadPolicyFile("xmlsocket://192.168.1.7:8082");
```

Siempre habría que llamar a este método `Security.loadPolicyFile()` antes de llamar al método `Socket.connect()`; es decir, antes de intentar establecer la conexión con el *socket*. *Flash Player* espera entonces hasta completar la solicitud del archivo de política antes de decidir si permite o no la conexión.

Sin embargo, si el *archivo maestro de política* especifica que la ubicación de destino no puede servir archivos de política, la llamada a `loadPolicyFile()` no será efectiva, incluso si hay un archivo de política en dicha ubicación.

Así, otro requisito de *Flash Player*, es que si se implementa un *servidor de socket* y se necesita proporcionar un *archivo de política de socket*, se debe decidir entre las tres posibilidades para proporcionar el *archivo de política de socket*:

- Puerto TCP 843
- Mismo puerto que el *socket* que acepta conexiones
- Utilizar otro puerto haciendo uso de `Security.loadPolicyFile()`.

En cualquier caso, para poder enviar una respuesta, el servidor deberá esperar a la primera transmisión de su cliente. Además, cuando *Flash Player* solicita un archivo de política, siempre transmite la siguiente cadena en cuanto se establece una conexión:

```
<policy-file-request/>
```

Cuando el *ESCC* recibe esta cadena en el puerto donde se decide que se proporciona el servidor del *archivo de política de socket* (puerto *TCP 843* como decisión de diseño), se transmite el *archivo de política socket*. La petición de *Flash Player* siempre termina con un *byte null* y la respuesta del servidor debe terminar con otro *byte* del mismo tipo (en nuestro caso nos aseguramos que el *archivo de política de socket* finalice con los caracteres “\n\0”).

Otra característica de la provisión de *archivos de política de socket* en *Flash Player* es que no se puede utilizar la misma conexión para la solicitud simultánea del *archivo de política de socket* y la solicitud de conexión hacia un determinado puerto. Por ello, el *ESCC* debería cerrar la conexión, una vez ha terminado de enviar el *archivo de política de socket*, en el caso de que se hubiera implementado el servidor de archivo de política sobre el mismo puerto que atiende las conexiones y que es el *8081* en el sistema implementado. En nuestra implementación esto no tiene efecto ya que la política se aprovisiona desde el puerto *TCP 843*; no obstante, se ha de tener en cuenta ante futuras modificaciones.

Adobe provee un servidor de *archivos de política de socket* en diferentes lenguajes de programación (*PERL* y *PYTHON*) [105], muy convenientes para el sistema implementado debido a las aplicaciones (*XAMPP*) que se han instalado en el servidor incluye *perl*; sin embargo, debido a que se ha programado gran parte del sistema en el lenguaje *php* se decide que éste es el lenguaje apropiado para proveer el *archivo de políticas socket*. Así, se decide realizar la provisión desde un *script* programado en *php*. Se podría haber escrito un *script* programando un *servidor socket php* que acepte conexiones al *puerto 843*, de la misma forma que se ha programado el *script* de *servidor socket* que gestiona las conexiones con los *ERCs*, sin más que tener en cuenta los comportamientos obligados que se han comentado en los párrafos anteriores. Sin embargo, ya hay disponible un servidor de *archivo de políticas de socket* para *Flash Player* programado mediante un *script php*, el cual tiene programadas algunas funcionalidades extra como es un sistema bastante completo de *log*. Este sistema *log* resultó muy útil a la hora de establecer un mecanismo de resolución de problemas, por lo que se adoptó, dentro del proyecto, esta solución. El código desarrollado, ligeramente adaptado a las necesidades del proyecto, se encuentra publicado bajo licencia *Creative Commons Attribution 3.0* y se puede consultar en la página oficial de su programador *Ammon Lauritzen* [109].

Con este *script* se consigue poder establecer conexiones *socket* en versiones superiores a *Flash Player 9.0.115* a partir de la descarga del *archivo de políticas socket* desde el puerto *TCP 843*. Esto es muy importante a la hora de evitar establecer dependencias con el *plug-in* de *Flash Player* instalado en el navegador ya que sin este servidor, el sistema funcionaría correctamente utilizando la *versión 9.0.115* de *Adobe Flash Player* e inferiores, pero no en versiones posteriores.

En este *script* se configura, por tanto, la provisión del archivo de políticas de *socket*, el cual se establece el siguiente:

```
$default_xml =
    '<?.?xml version="1.0" encoding="UTF-8"?'.>'.
    '<cross-domain-policy
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```



```
xsi:noNamespaceSchemaLocation="http://www.adobe.com/xml/schemas/PolicyFileSocket.xsd">'.
    '<allow-access-from domain="*" to-ports="*" secure="false" />'.
    '<site-control permitted-cross-domain-policies="all" />'.
'</cross-domain-policy>';
```

Se ha de notar que éste *archivo de políticas de socket de Adobe Flash*, se recibe tanto en las *aplicaciones swf* que acceden al *servidor socket* desde la *Interfaz Gráfica de Usuario* desde el *EACCC*, como desde las aplicaciones embebidas en la presentación del contenido que controlan el *socket* en los *ERCs*: *CONTROL SWF* (ver apdo. 5.4.4.4.3.2 Aplicación Control swf y Clientes Socket en los *ERCs*).

5.4.3.4.3 Servidor de Contenido Multimedia: Arquitectura Auto-Proxy

Una funcionalidad muy importante dentro del *Marco* ha sido la de desarrollar dentro de él una *Arquitectura Auto-Proxy* según se ha definido ésta en el contexto del proyecto, que define la interacción entre el *ESCC* y los *ERCs* para la funcionalidad de provisión de contenidos multimedia a reproducir en los *ERCs*.

Una de las funciones principales del *ESCC* es la de proveer ese contenido multimedia a los *ERCs*, para ser reproducido sin necesidad de que los *ERCs* tengan que realizar nuevas peticiones del contenido; de esta forma, es posible optimizar en muchos casos el *ancho de banda* desde el *ESCC* hacia los *ERCs*.

En la versión inicial del sistema, los *ERCs* realizaban peticiones para descargar flujos *Pseudo Streaming (HTTP Streaming)* directamente del *ESCC*, en los instantes en que la *lista de reproducción* programada (para cada *Zona*, de cada *Escenario* de cada equipo), indicaba que había que mostrar cualquiera de los archivos de contenido multimedia. Sin embargo, este modo de funcionamiento provocaba que se utilizara un *ancho de banda* elevado en el sistema, el cual crecía según lo hacía el número de *ERCs* (o número de *Zonas* en la configuración) de la arquitectura. Por este motivo, se decidió idear un nuevo mecanismo de reproducción de contenido multimedia en el que los archivos sean reproducidos de forma local directamente desde cada *ERC*. Para ver en detalle la problemática del consumo de *ancho de banda* se puede consultar el capítulo de pruebas (ver Resultados de las Pruebas Realizadas), en el cual se ha forzado el *Marco* a funcionar sin la *Arquitectura Auto-Proxy*.

El *ESCC* es un servidor de contenido multimedia e inicialmente es donde se suben los archivos multimedia que serán presentados en cada *Sistema de Presentación de Contenido*. Posteriormente ese contenido multimedia es distribuido a cada uno de los *ERCs* desde el *ESCC* mediante la aplicación *rsync*. Así, la configuración del *servidor de contenido* se ha complementado con el servicio *rsync* ejecutándose desde inicio (ver apdo. 5.4.3.3.3 Configuraciones para la Sincronización del Contenido Multimedia: Servidor *rsync*).

Una vez que el contenido se descarga en los *ERC*, el reproductor de contenido multimedia (*repZona.swf*) establecerá la lógica apropiada para reproducir el contenido desde el propio *ERC (localhost)*. Para que las aplicaciones *swf* puedan interactuar mediante peticiones a *localhost* deben descargar inicialmente un *archivo de política url* (ver ANEXO E).

5.4.4 Configuraciones y Aplicaciones en los ERCs para Proveer la Reproducción de los Archivos

Este subapartado presenta el funcionamiento de los ERCs en el *Marco*. De igual modo que para el ESCC, se va a presentar dos aspectos importantes:

1. Se mostrarán las configuraciones y desarrollos, de los elementos más significativos, establecidas en este equipo para el *Marco* diseñado para establecer la configuración.
2. Se presentará el funcionamiento específico de cada uno de los elementos más representativos desarrollados en este equipo, para el *Marco* diseñado para proporcionar el funcionamiento lógico.

En el ESCC, el elemento más definitorio del equipo es la provisión de una *Interfaz Gráfica de Usuario* que establece la configuración del *Marco*. En el caso del ERC el objetivo último es, por una parte, recuperar la configuración y contenido del ESCC y, por otra parte, establecer un reproductor de acuerdo a la configuración recuperada. Por ello, el elemento que define el núcleo del ERC es la aplicación que recupera la configuración y reproduce el contenido multimedia para la aplicación de *Cartelería Digital*.

La *Difusión Selectiva*, en el contexto del proyecto, se establece a través de la *dirección IPv4* como parámetro diferenciador o clasificador. Esta lógica de funcionamiento se establece en el ESCC como se ha explicado. Sin embargo, es necesario almacenar y poder configurar diversa información en el propio ERC; por tanto, existe otra parte importante también que es la de proveer una *Interfaz Gráfica de Usuario* para establecer ciertos parámetros del sistema en el ERC. Sin embargo, en comparación con la configuración del ESCC, la de un ERC es menos sofisticada ya que no se hace a través de peticiones *asíncronas* ni de *aplicaciones swf*.

Se ha utilizado para el servidor web del equipo ERC, al igual que en el ESCC, el conjunto de aplicaciones recopiladas en XAMPP en su versión 1.7.4. Este conjunto de aplicaciones nos permite poder configurar el ERC de forma remota a través de un navegador web. Entre otros parámetros, los más importantes que se permitirá configurar son las *direcciones de IPv4*, junto con la *máscara* y el *router por defecto* de las interfaces de red. De este modo se proporciona conectividad y se aprovisiona al equipo cuál es la *dirección IPv4* del ESCC. También se podrá configurar la cantidad de *ancho de banda* a utilizar por la aplicación `rsync` en cada ERC.

El ERC presentará el contenido a través de un navegador web a pantalla completa y hará uso de aplicaciones *swf* para presentar los videos y las imágenes, además de interpretar las órdenes desde el ESCC mediante una monitorización de la conexión al *servidor socket*. Es por ello que será necesario tener instalado el *plugin Flash Player* en el navegador utilizado, que en el proyecto será *Mozilla Firefox* por ser una distribución libre y compatible con el sistema operativo *Linux* entre otras razones.

5.4.4.1 Sistema Operativo de los ERCs

El *sistema operativo* finalmente utilizado en los ERCs ha sido *Ubuntu Desktop 10.10*. Se ha utilizado este sistema para proporcionar, por una parte, un *Interfaz Gráfica de Usuario* a través de un *servidor web* funcionando en el propio equipo y, por otra, realizar una representación del contenido multimedia que se ha configurado en el ESCC. La configuración, proporcionada por el ESCC, se aplica para la *Ubicación* a la que corresponde la *dirección IPv4* que se configura en el ERC y es necesario utilizar la salida gráfica del equipo, para proporcionar la presentación de los videos.

5.4.4.2 Secuencia de Configuración en el Inicio de los ERCs

Antes de entrar en detalle acerca de las aplicaciones y configuraciones necesarias, se debe tener una visión general del funcionamiento de los *ERCs*. Éste se resume, en parte, en este sub-apartado, de modo que se muestra qué configuraciones se realizan una vez se inicia el equipo. La secuencia específica del *ERC*, se establece en el archivo `inicioCliente.php` y se establece en el *Nivel de Ejecución 2* del *sistema operativo*, ya que es éste el nivel de ejecución por defecto de *Ubuntu*:

```
pfc@erc:~$ ln -s /opt/lampp/htdocs/div-dat/div-client/INICIO/inicioCliente.php /etc/rc2.d/S95inicioCliente.php
```

Al iniciar el sistema *ERC* se ejecutan una serie de aplicaciones y *script* para configurar el *sistema operativo* a partir de datos almacenados en la *base de datos*. Estas configuraciones almacenadas en las tablas de la *base de datos* las introduce el administrador del sistema a través de la *Interfaz Gráfica de Usuario* desarrollada mediante el uso de herramientas web.

Además de la configuración, común al *ESCC*, relativa a las interfaces de red (ver apdo. 5.4.2.2). A través del *script* `inicioCliente.php`; aplicamos las siguientes instrucciones adicionales:

1. Se realiza un sincronismo de hora al inicio con respecto al servidor *ESCC*, ya que este será el sistema de referencia de hora (ver apdo. 5.4.4.3.1 Configuraciones para la Sincronización de Hora en el *ERC*).

```
$salida.=shell_exec("sudo ntpdate-debian");
```

2. Se realiza una sincronización, del contenido multimedia de los archivos que se han subido al *ESCC*, con el comando `rsync` cada vez que se inicia el equipo (ver apdo. 5.4.4.3.4 Configuración del Cliente `rsync`).

```
$archivo=fopen('archivoRsync.txt','r+');

if($archivo){
    while (!feof($archivo)) {
        $salida.=fgets($archivo, 255);
    }
    fclose ($archivo);
    system("$salida");
}
```

Con estas instrucciones se indica al sistema que ejecute la orden almacenada en `archivoRsync.txt`. En este archivo (ver apdo. 5.4.4.3.4) se indica que sincronice el contenido multimedia utilizando un *ancho de banda* determinado y que la conexión se realice con el contexto *multimedia* del *servidor rsync* el cual se aprovisiona en el *ESCC* en el archivo `/etc/rsync.conf` con la aplicación *rsync* ejecutándose como servicio (ver apdo. 5.4.3.3.3 Configuraciones para la Sincronización del Contenido Multimedia: Servidor *rsync*).

Una vez se ha iniciado el sistema con las *interfaces de red* configuradas correctamente, la *hora establecida* y se ha realizado la petición de *sincronismo de contenido multimedia*, el sistema puede comenzar a realizar peticiones de la configuración al servidor para emitir un determinado *Escenario* de la *Ubicación* a la que pertenece el equipo.

Así, una vez se ha iniciado el sistema, se lanza la aplicación *Firefox* a pantalla completa, como veremos (en el apdo. 5.4.4.3.2), con el *plugin R-kiosk* y otras aplicaciones previamente instaladas. De este modo, el *ERC* presentará al iniciar el sistema un navegador web a pantalla completa realizando inicialmente una petición de una página web local. En los sucesivos apartados se muestran las configuraciones realizadas.

5.4.4.3 Configuraciones Específicas de los ERCs

Una de las características que se ha considerado a la hora de abordar el diseño del *Marco* ha sido la de establecer, en parte, independencia del *sistema operativo* que soporta los distintos equipos. Sin embargo, debido a que finalmente se ha abordado el problema utilizando el *sistema operativo Linux* (por algunas ventajas que aporta en una situación de comercialización potencial del sistema), se muestran algunos aspectos que se han configurado y que, aunque existen otras alternativas de desarrollo, se han elegido éstas por su sencillez y rapidez de implementación.

Algunas de las aplicaciones usadas, posiblemente pudieran ser mejoradas en términos de eficiencia en el uso de memoria o procesado; no obstante, un desarrollo a medida sin utilizar aplicaciones de uso extendido (navegadores web), hubiera retrasado inevitablemente los tiempos del proyecto (sobre toda en el desarrollo de la aplicación que presenta los videos).

En los siguientes subapartados se detallan las configuraciones realizadas en cada uno de los elementos fundamentales que forman el *Marco* final.

5.4.4.3.1 Configuraciones para la Sincronización de Hora en el ERC

Cada vez que se cambia el servidor *ESCC* en la configuración del *ERC*, se debe ejecutar la sincronización de hora a través del cliente *ntp* del *ERC* (*ntpdate*). Por tanto, no es necesario instalar *ntpd* en el *ERC* como servicio, sino ejecutar simplemente *ntpdate* como cliente. Para verificar qué está instalado relacionado con *ntp*, en *Ubuntu* se puede ejecutar el siguiente comando para verificar la instalación de la aplicación:

```
pfc@erc:/bin$ dpkg -l | grep ntp
ii ntpdate          1:4.2.4p8+dfsg-1ubuntu6
Client for setting system time from NTP serv
```

Cuando se ejecuta *ntpdate* se indica que actualice la sincronización de hora cuando se ejecuta el comando. Por comodidad, se ejecuta la aplicación *ntpdate-debian*, a la cual no hay que pasarle ningún parámetro ya que utiliza los almacenados en */etc/defaults/ntpdate* y el funcionamiento es idéntico a *ntpdate*. Por tanto, en el *script* de inicio se ejecuta:

```
$salida.=shell_exec("sudo ntpdate-debian");
```

Al ejecutar *ntpdate-debian* no es necesario pasarle parámetros, ya que estos lo almacena en */etc/defaults/ntpdate*, para este archivo se tiene configurado:

```
# The settings in this file are used by the program ntpdate-debian, but not
# by the upstream program ntpdate.
# Set to "yes" to take the server list from /etc/ntp.conf, from package ntp,
# so you only have to keep it in one place.
NTPDATE_USE_NTP_CONF=yes

# List of NTP servers to use (Separate multiple servers with spaces.)
# Not used if NTPDATE_USE_NTP_CONF is yes.
NTPSERVERS="no.aplica.es"
```

```
# Additional options to pass to ntpdate
NTPOPTIONS=""
```

De este modo, la configuración se obtiene finalmente del archivo `/etc/ntp.conf`, aunque no esté instalado el servicio `ntpd`. Esta configuración se establece de forma sencilla en función de la configuración del *ERC*:

```
pfc@erc:~$ cat /etc/ntp.conf
servers <IP_ESCC>
```

La modificación de este archivo se realiza cada vez que modificamos la configuración de la dirección *IPV4* del *ESCC* a través de la *Interfaz Gráfica de Usuario*. Como se puede apreciar la configuración es muy simple y similar a la del *ESCC*, por otra parte, en los *ERCs* la referencia de hora es el *ESCC*.

5.4.4.3.2 Configuración para Ejecución de Aplicaciones Adicionales al Inicio del Sistema

Además de la secuencia de arranque descrita en el apartado 5.4.4.2 (Secuencia de Configuración en el Inicio de los *ERCs*), se deben lanzar otras aplicaciones cuando el sistema se inicia. Estas aplicaciones están relacionadas con la manera de presentar el contenido multimedia en el *ERC* mediante el *SPC* (*Sistema de Presentación de Contenido*).

Se configura el sistema para que al inicio de la sesión se ejecute el navegador web *Mozilla Firefox*. Como parámetro principal al navegador se añadirá la página web de inicio que se desea cargar (ver detalle de la página de inicio sin conexión en el apdo. 5.4.4.4.2). Esta página de inicio será una página escrita en *php* y se almacena de forma local²⁴ de modo que realizará una serie de comprobaciones antes de redirigir las peticiones al *ESCC*.

Según se instale un *sistema operativo linux* u otro, se tendrá una determinada interfaz gráfica que controle el sistema: *Gnome*, *XFCE* o *KDE*. La configuración se realiza de forma diferente, aunque en todos los casos es sencilla. Detallamos la configuración en *Gnome* (*gdm 2.32*) ya que es la interfaz gráfica de la distribución de *Linux* (*Ubuntu Desktop 10.10*) utilizada para la implementación del *ERC*:

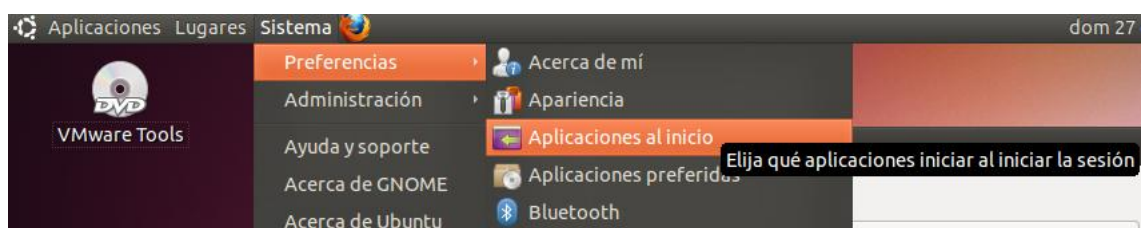


Ilustración 39.-. Menú de configuración de las aplicaciones de inicio sobre Ubuntu 10. 10 basado en Gnome.

Una vez que pulsamos sobre este elemento del menú (*Sistema>>Preferencias>>Aplicaciones de Inicio*) nos aparece una ventana emergente donde se permite configurar diferentes aplicaciones de inicio adicionales:

²⁴ Se ha de recordar que los *ERC* también incorporan un servidor web a través del conjunto de aplicaciones *XAMPP*.

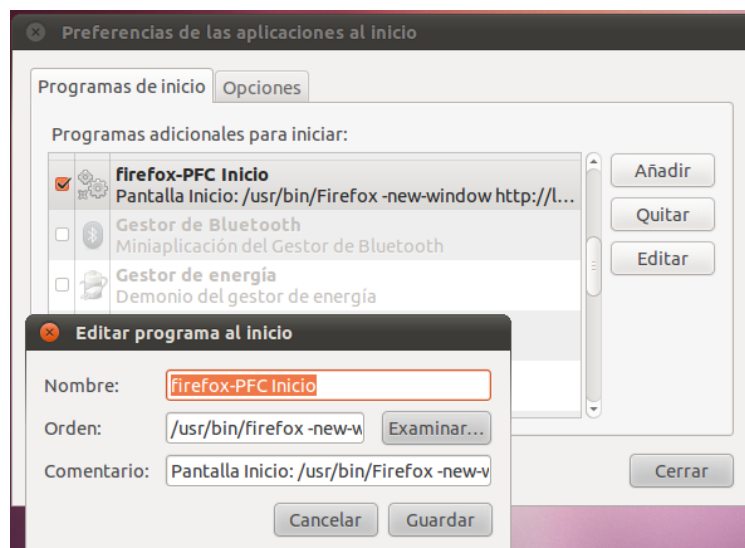


Ilustración 40.-. Configuración de Mozilla Firefox al inicio del ERC.

La aplicación que configuramos para que se ejecute desde el inicio es *Mozilla Firefox* y la página de inicio se genera con el *script* `pantallaInicio.php`:

```
/usr/bin/firefox      -new-window      http://localhost/div-dat/div-
client/INICIO/pantallaInicio.php
```

Se ha de considerar que además de ejecutar *Mozilla Firefox* al inicio del sistema, también se deben desactivar algunas aplicaciones como, por ejemplo, la *gestión de energía* para que la pantalla no se desactive al no haber actividad aparente en el sistema. También se desactivan otras aplicaciones como la *notificación de actualizaciones del sistema* y otras aplicaciones que no se utilizan (*Network Manager*, *Buscar Nuevos Controladores Hardware*, *Bluetooth Manager*, *Power Manager*, *Update Notifier*, *Print Queue Applet*,...).

El resumen del funcionamiento lógico del *script* `pantallaInicio.php`, se detalla en subsiguientes apartados donde se dan explicaciones más precisas de este *script* (ver apdo. 5.4.4.4.2 Página de Inicio en el Sistema de Presentación de Contenido (SPC): Pantalla Inicial Sin Conexión).

5.4.4.3.3 Configuraciones de la Aplicación Firefox y Otras Relacionadas

Debido a que el sistema se puede reiniciar a través de la interfaz web (ver APENDICE C.II.2 MENÚ SISTEMA) o ante posibles reinicios inesperados del sistema es, también, destacable indicar que hay que realizar ciertas configuraciones sobre el navegador web para que, por ejemplo, no se cargue la última página que se estaba mostrando. Si no se realizaran estas configuraciones cada vez que se reinicia el sistema por algún motivo, además de lanzar la página de inicio configurada, se mantienen en otras pestañas las peticiones hacia el *ESCC* que se hubieran establecido en las sesiones anteriores.

Una forma de modificar los parámetros de configuración del navegador es tecleando **about: config** directamente en la barra de direcciones del navegador. Una vez se accede a los parámetros, desactivamos el restablecimiento de la sesión anterior:

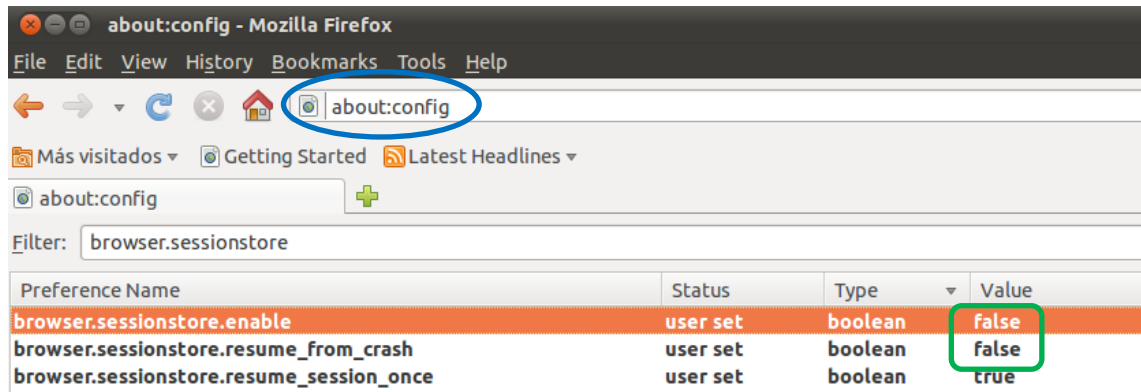


Ilustración 41.- Configuración del navegador Mozilla Firefox para que no inicie la última sesión.

En la Ilustración 41 se muestran los valores de configuración que se han de modificar. Se ha de notar que el navegador *Firefox* dispone de una gran cantidad de parámetros y que se ha aplicado un filtro para visualizar los que realmente interesan dentro de nuestro *Marco* implementado (*browser.sessionstore*).

Como la presentación del contenido multimedia se realiza mediante el navegador web y un *SPC* (*Sistema de Presentación de Contenido*), se debe lanzar éste (en nuestro caso *Mozilla Firefox*) a pantalla completa y se debe también deshabilitar el ratón sobre la pantalla para que éste no se visualice. Existen varias alternativas para desactivar el ratón (y/o teclado). Por ejemplo, para desactivar el ratón (y teclado) se puede hacer a través de la configuración del archivo `/etc/X11/xorg.conf`; no obstante, se hace uso de una aplicación llamada *unclutter* que deshabilita el ratón si no se está usando durante un tiempo determinado.

Para instalar esta aplicación basta con disponer de una conexión a *Internet* bajo los repositorios por defecto del sistema:

```
pfc@erc:~$ sudo apt-get install unclutter
[...]
pfc@erc:~$ dpkg -l | grep unclutter
ii unclutter      8-13
    hides the cursor in X after a period of inactivity
```

Esta aplicación deshabilita el ratón por defecto a los 5 segundos. Pudiendo ser este parámetro modificable (ver `man unclutter`).

Para establecer el navegador a pantalla completa se hace uso de una extensión del navegador: el *Add-ons R-kiosk* (*Real Kiosk*) para *Firefox* [110]:

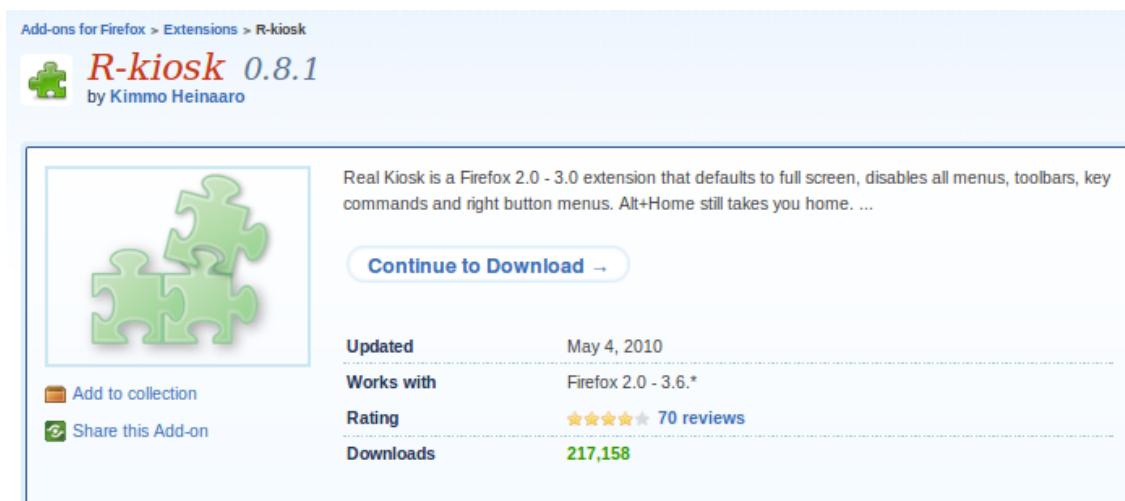


Ilustración 42.-. Add-ons para Mozilla Firefox que permite mostrar el contenido del navegador a pantalla completa.

Además, esta extensión de *firefox* permite, entre otras funcionalidades, deshabilitar también el acceso al menú de *firefox* por lo que el sistema queda protegido ante conexiones de teclados y otros periféricos con los que se pretendiese acceder a la configuración del sistema.

5.4.4.3.4 Configuración del Cliente *rsync*

Es necesario descargar el contenido multimedia desde el *ESCC* en cada *ERC*, para disponer de él localmente en cada *ERC* para su reproducción (arquitectura *Auto-Proxy*) y para ello se utiliza en el *Marco*, como ya se ha comentado, la aplicación *rsync*.

Para ello, es necesario que se ejecute *rsync* como servicio en el *ESCC* (ver apdo. 5.4.3.4.3 Servidor de Contenido Multimedia: Arquitectura *Auto-Proxy*) y se mantenga activo a modo de servicio al iniciar este equipo. Para proveer la sincronización, se ha programado lanzar la aplicación en modo cliente en cada *ERCs*; de esta manera, se conecta al *ESCC* a través de esta aplicación y se descargan o eliminan los archivos de forma local, igualándolos a los que haya en el servidor (*ESCC*). De esta forma, no es necesario realizar peticiones del archivo a través de la *WAN* cada vez que se tenga que reproducir un archivo programado en la lista, debido a que se tiene almacenado una copia de forma local (ver apdo. 4.3.2 Arquitectura Funcional *Auto-Proxy* del Capítulo 4). Con esta estrategia, el archivo se descarga una sola vez, manteniendo sincronizado el contenido multimedia en el *ERC* con el que se ha cargado en el *ESCC*.

Se ha planteado realizar la sincronización como una tarea programada que se debería ejecutar cada cierto tiempo con la herramienta *cron*. Esta configuración es posible obtenerla editando el fichero `/etc/crontab` en el cual se añade la siguiente línea:

```
# m h dom mon dow   command
30 * * * * root /opt/lampp/htdocs/div-dat/div-client/rsyncCronEjec.sh
```

La configuración anterior permite ejecutar con el script `rsyncCronEjec.sh` cada 30 minutos.

Además de esta configuración, se complementa con una estrategia que ejecuta el *cliente rsync* no solo a intervalos pre-programados, sino que se ejecuta también cuando se detecta un

cambio en los archivos presentes en el servidor. Estos cambios se informan a través del *socket*. Así, cuando se sube o elimina un archivo al *ESCC*, éste escribe sobre el *socket* una orden indicando, al resto de *ERCs*, que deben establecer una sincronización de archivos con el servidor.

Una vez que el *ERC* detecta que debe aplicar la sincronización, por medio de la conexión al *servidor socket*, ejecuta la aplicación *rsync* con los parámetros almacenados en la *base de datos* del *ERC* (*ancho de banda de sincronización* y *dirección IPv4* del *ESCC*). Por comodidad, cada vez que se detecta la orden en el *servidor socket*, se ejecuta la orden del sistema almacenada en *archivoRsync.txt* la cual invoca a la aplicación *rsync* para la sincronización del contenido desde el *ESCC* al *ERC*. La orden almacenada en este archivo también se ejecuta al inicio del *ERC* como se ha indicado en el apartado 5.4.4.2 Secuencia de Configuración en el Inicio de los *ERCs*.

La edición de los archivos *rsyncCronEjec.sh* y *archivoRsync.txt*, donde se almacena la configuración de la aplicación cliente *rsync*, se realiza mediante los *script php* que modifican, en la *base de datos*, dos parámetros del *ERC*:

- la *dirección IPv4* del servidor donde se ejecuta el servicio del servidor *rsync*; es decir, la *dirección IPv4* del *ESCC*.
- el *ancho de banda* a utilizar entre el *ESCC* y el *ERC* para actualizar los archivos.

Estos parámetros se configuran a través de la *interfaz gráfica del usuario* administrador del sistema. Así, cuando se editen alguno de estos parámetros se producirán las siguientes acciones en el *ERC*:

- Se actualizarán los valores en la *base de datos*
- Se editará los archivos *rsyncCronEjec.sh* y *archivoRsync.txt*
- Se aplicará la orden de sincronización de contenido multimedia

Un ejemplo, del contenido de los archivos generado con la configuración de la sincronización, es el siguiente:

```
rsync -av --bwlimit=128 --delete pfc@192.168.103.7::multimedia
/opt/lampp/htdocs/div-dat/div-client/archivosLocales/ > /dev/null
```

En el ejemplo, el *ancho de banda* establecido son 128 KB/s y que el equipo servidor del contenido multimedia (*ESCC*) es el 192.168.103.7.

Un aspecto interesante de esta configuración es la posibilidad de establecer los equipos servidores de contenido multimedia mediante el protocolo *rsync* en un esquema con forma de árbol. De este modo, se podría configurar un solo equipo en cada sede remota para que haga uso del enlace *WAN* para la sincronización del contenido multimedia y configurar el resto de equipos para que actualice su contenido con respecto al que hace uso del enlace *WAN*. No obstante, esta mejora se mantiene como línea de trabajo futuro.

5.4.4.4 **Lógica Funcional del ERC en el Marco de Aplicaciones de Difusión Selectiva Diseñado y Desarrollado**

La funcionalidad del *ERC* dentro del *Marco* se centra en la parte de la *arquitectura funcional* en la que éste obtiene la configuración establecida en el servidor y, a partir de ésta, difunde el contenido multimedia programado para dicha configuración. De este modo, el desarrollo del *ERC* se centra:

- Definir una *interfaz gráfica* para así poder proporcionar al *ERC* la información necesaria para que pueda conectarse con el *ESCC*
- Establecer los mecanismos de comunicación con los que el *ERC* se comunicará con el *ESCC*

La generación de la *Interfaz Gráfica de Usuario* implica varios aspectos a tener en cuenta. Uno de ellos es el modelo que se establecerá para generar las páginas de configuración (*html*) y otro, la lógica que se aplicará cuando se realicen estas configuraciones. La interacción que se establecerá con el *ESCC*, a partir de la propia *lógica* establecida en el *ERC*, se realizará, por tanto, en función de la configuración que se esté llevando a cabo en el *ERC*.

Las interacciones con el *ESCC* dependerán de la funcionalidad que tenga objeto. Es decir, se establecerán diversos mecanismos de comunicación desde el *ERC* con el *ESCC*, según se quiera recuperar la configuración (*http*), obtener los archivos multimedia para almacenarlos de forma local (arquitectura *Auto-Proxy* mediante `rsync`) o reconocer las órdenes enviadas por el *ESCC* (conexión *socket*).

En los siguientes sub-apartados se explica cómo se implementa la *Interfaz Gráfica de Usuario* y los mecanismos y aplicaciones desarrolladas para obtener los objetivos marcados en el *ERC* para la presentación de contenido multimedia.

5.4.4.4.1 *Plataforma de Configuración del Marco en los ERCs*

La plataforma de configuración de los *ERCs* sigue el mismo esquema (Ilustración 35) que se ha presentado en el apartado 5.4.2.3.1 Secuencia Lógica de Funcionamiento de la Aplicación Web de Configuración (Interfaz Gráfica de Usuario). Sin embargo, en la configuración de estos equipos, no se hace uso de aplicaciones *swf* embebidas en el código *html* como se aprecia en dicha ilustración.

La razón es que en estos equipos no se hace uso de peticiones *asíncronas* de datos, ya que realizar cambios en la *dirección IPv4* del *ERC*, provocaban que se perdiera la conectividad con el equipo. No obstante, se ha ideado un mecanismo que permite no perder dicha conectividad, manteniendo la *dirección IPv4* anterior hasta que se establezca el reinicio del servidor. Como además las transacciones serán muy puntuales y apenas generan tráfico, se realiza un esquema más sencillo sin peticiones *asíncronas*.

Como se observa en la Ilustración 35 (pág. 127), en el *ERC* se hace uso de formularios de código *html* y se navega de forma automática a otras páginas *html*. Como norma general, en la configuración de cada página y ejecución de formularios se navegará a la misma página (*php*), pero con una serie de parámetros informados (los datos enviados por el formulario). De este modo se generará de nuevo código *html* en función de los parámetros recibidos.

La secuencia del funcionamiento de la plataforma de configuración del *ERC* se muestra en el siguiente esquema:

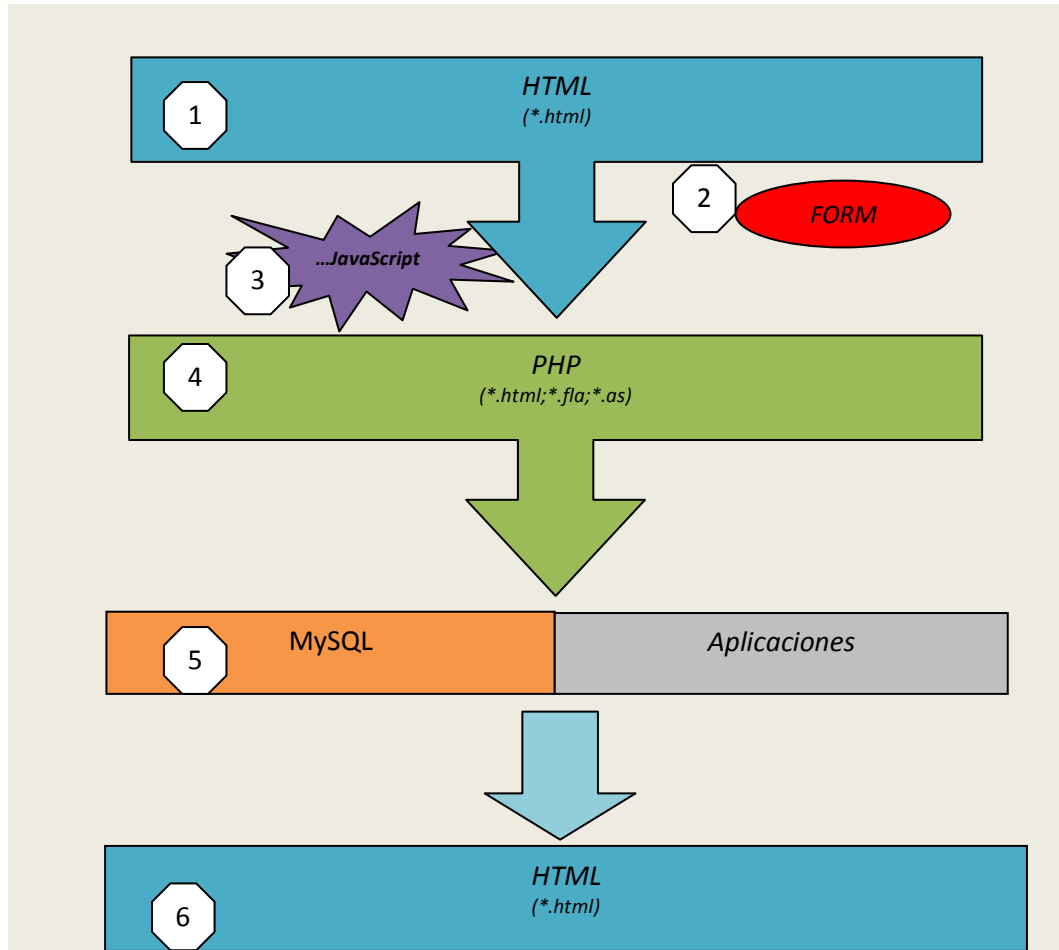


Ilustración 43.- Esquema general de funcionamiento modular de la Arquitectura Software de configuración desarrollada en el ERC.

La página en *html*, para generar la configuración de los *ERCs*, se establece de forma clásica en un modelo de programación en el que se devuelve código *html* de forma dinámica en función de la petición a un *script php*. En estas páginas, debido a la sencillez, no se implementa ningún tipo de *Modelo Vista-Controlador*, ni se hace uso de peticiones *asíncronas*; sin embargo, establecen comunicación con la *base de datos* (ver *Config LOCAL PHP* en Ilustración 35) y ejecutan aplicaciones que permiten obtener los resultados esperados (*ntpdate-debian*, *rsync*). Se ha de indicar que para establecer la configuración se utilizan formularios *html* (ver *HTML* en Ilustración 35) que vuelven a llamar al mismo *script php*, enviando la información mediante métodos propios del protocolo *http*: GET/POST; y así, según la información recibida, el *script php* generará un código *html* u otro.

HTML y Config. LOCAL en los ERCs

La *Interfaz Gráfica de Usuario* de configuración para los *ERCs* basa gran parte de su lógica en un mismo patrón. Este patrón de configuración permite establecer los parámetros de configuración necesarios (*dirección IPv4* del *ERC*, *máscara*, *gateway por defecto*, *ancho banda sincronización*, *dirección IPv4* del *ESCC*) y almacenarlos en la *base de datos* local y, además, permite ejecutar algunas aplicaciones (*ntpdate-debian*, *rsync*, *ifconfig*, *ping*, ...) sobre el *ERC* desde determinadas páginas web.

El esquema de configuración se basa en la cumplimentación de formularios por lo que no se ha implementado ningún mecanismo basado en eventos.

A continuación se muestra un esquema del lógico general interno de la *interfaz gráfica* implementada para la configuración del *ERC*:

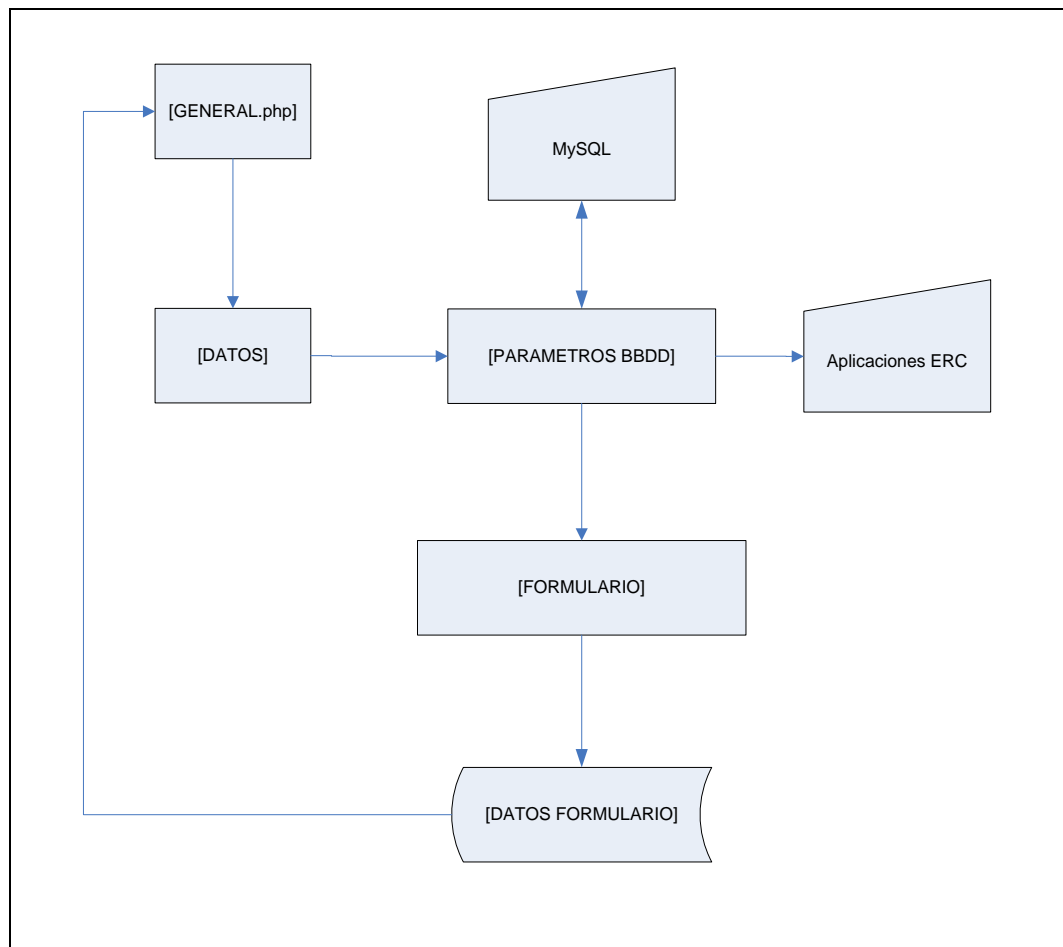


Ilustración 44.-. Esquema de funcionamiento del interfaz gráfico de usuario

Al comenzar la ejecución del *script* recoge los datos que se le han enviado y, a partir de estos, recupera/escribe los parámetros en la *base de datos*. Según sea la página podría invocar a aplicaciones en el *ERC* o simplemente presentar el formulario que permite al usuario, a través de una interfaz, modificar los parámetros del *ERC* (*dirección IPv4 del ERC, máscara, gateway por defecto, dirección IPv4 del ESCC, ancho de banda para sincronización*). Una vez el usuario decide enviar los nuevos parámetros se vuelve a llamar a la propia página con los nuevos datos del formulario.

A continuación se presenta una tabla con un listado y un breve resumen de los archivos que forman la *interfaz gráfica de usuario* en el *ERC*:

ERC		
Script PHP	Parámetros Accedidos	Acciones: Descripción
DAT_SERV\IPServ.php	IP_Serv IP_Back BW_rsync	<ul style="list-style-type: none"> Modifica la <i>dirección IPv4</i> del ESCC. Ejecuta la sincronización de hora (<i>ntpdate-debian</i>) contra el servidor configurado. Permite lanzar otras aplicaciones en el cliente: <i>PINGServ.php</i>
DAT_SERV\PINGServ.php	---	<ul style="list-style-type: none"> Ejecuta la aplicación PING contra el ESCC y servidor configurado como backup.
DAT_CLIENT\IPCliente.php	IP_Cliente	<ul style="list-style-type: none"> Modifica los parámetros básicos de red del ERC en la

ERC		
Script PHP	Parámetros Accedidos	Acciones: Descripción
	MASK Gateway	<i>base de datos.</i> <ul style="list-style-type: none"> Modifica los parámetros de red básicos en el <i>ERC</i> a nivel de sistema. Posteriormente navega a la página que muestra el estado de las <i>direcciones IPv4</i> de las interfaces del <i>ERC</i> (<i>StatusCambioIP.php</i>)
DAT_CLIENT\StatusCambioIP.php	---	<ul style="list-style-type: none"> Presenta el resultado de ejecutar algunas aplicaciones del sistema: <i>ifconfig</i>, <i>cat /etc/resolv.conf</i>, <i>route</i>, <i>ping</i>,...
DAT_CLIENT\AnchoBandaSincro.php	BW_rsync	<ul style="list-style-type: none"> Permite cambiar el <i>ancho de banda</i> utilizado para sincronizar los archivos multimedia con el <i>ESCC</i>. Permite establecer la orden de sincronización (<i>archivoRsync.txt</i>). Esta será el método de sincronización al inicio del <i>ERC</i> con el <i>ESCC</i>. Ejecuta la orden de sistema almacenada (<i>archivoRsync.txt</i>)
SISTEMA\CambiaPassAdmin.php	Clave (Admin)	<ul style="list-style-type: none"> Permite cambiar la clave del usuario 'admin'.
SISTEMA\CambiaPassSistema.php	Clave (Sistema)	<ul style="list-style-type: none"> Permite cambiar la clave del usuario 'sistema'
SISTEMA\reiniciar.php	---	<ul style="list-style-type: none"> Permite reiniciar el <i>ERC</i>. Es útil cuando se ha cambiado de <i>dirección IPV4</i> en el <i>ERC</i> o se ha detectado algún problema. Ejecuta la orden de sistema <i>reboot</i>.
ESTADO>ListadoArchivos.php	----	<ul style="list-style-type: none"> Muestra un listado de los archivos almacenados de forma local. Permite la descarga de estos archivos a través de enlaces.
variables_comunes_estaticas_cliente.php	----	<ul style="list-style-type: none"> Recupera una serie de parámetros necesarios y fijos en el <i>ERC</i> (<i>claves y usuario de la base de datos</i>). Se incluye en todas las páginas de configuración (ver Ilustración 35).
menu.php	-----	<ul style="list-style-type: none"> Muestra el menú para todas las interfaces gráficas de usuarios en el <i>ERC</i>. Dependiendo del usuario tendrá una visibilidad u otra. Se incluye en todas las páginas de configuración (ver Ilustración 35).

Tabla 16.-. Resumen de archivos que conforman la Interfaz Gráfica de Usuario en el ERC.

En la tabla anterior se puede observar un resumen de los parámetros que se pueden modificar en el marco a través de la *Interfaz Gráfica de Usuario*. En el ANEXO C.II se muestra el manual de la *Interfaz Gráfica de Usuario Administrador* del ERC.

5.4.4.4.2 Página de Inicio en el Sistema de Presentación de Contenido (SPC): Pantalla Inicial Sin Conexión

La URL configurada en Mozilla Firefox cuando se inicia el sistema (ver apdo. 5.4.4.3.2 Configuración para Ejecución de Aplicaciones Adicionales al Inicio del Sistema) es un *script php* local (que se ejecuta mediante las aplicaciones del servidor web *XAMPP*) almacenado en el *ERC* (*pantallaInicio.php*) y que monitoriza si se tiene acceso a la *base de datos* del servidor (*ESCC*). Este código *html* generado se muestra a pantalla completa a través del *SPC* (*Sistema de Presentación de Contenido*), el cual, en nuestro contexto de pruebas, será la propia consola de la aplicación *VMWare* configurada para que se salda se presente en una pantalla: TFT, proyector, TV con entradas de video,...

El esquema lógico de esta página inicial (`pantallaInicio.php`) es el siguiente:

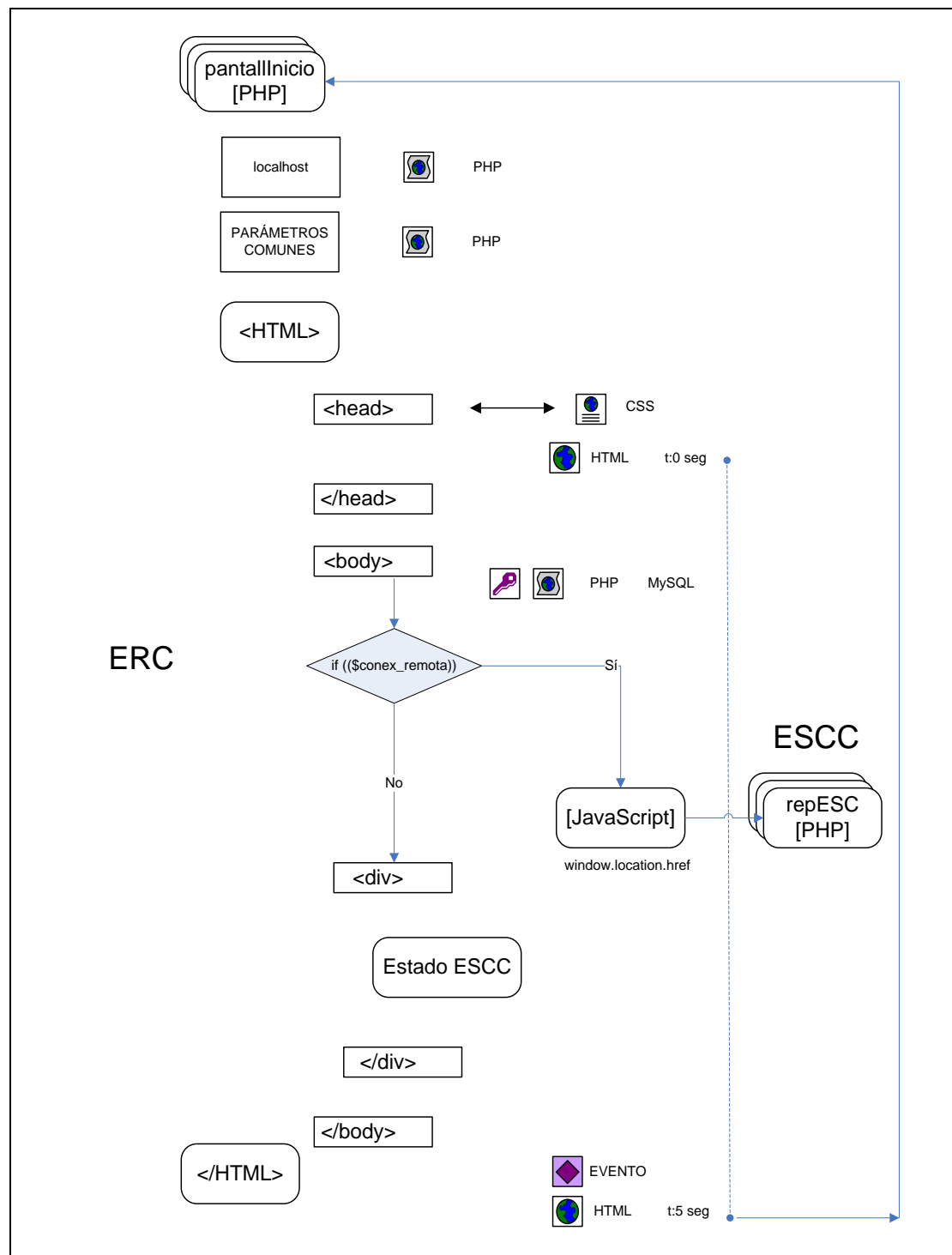


Ilustración 45.-. Esquema lógico de funcionamiento de la página de la pantalla de Inicio presentada por el SPC en cada ERC.

El intervalo de tiempo en el que se vuelve a recargar la página es el intervalo de tiempo para el cual se está evaluando la conectividad con el servidor. Este temporizador se ha establecido a 5 seg. En caso de poder conectar con la *base de datos* se redirigen las peticiones a la página que evalúa la *dirección IPv4* del ERC, la cual está alojada en el ESCC, de modo que devuelve la configuración para ese equipo (*Ubicación*).

La página local (*pantallaInicio.php*) presenta, a través del *SPC*, la siguiente imagen (código *html*) mientras no se detecte el funcionamiento correcto del *ESCC*; es decir, hasta que se detecte que se tiene acceso a la *base de datos*:

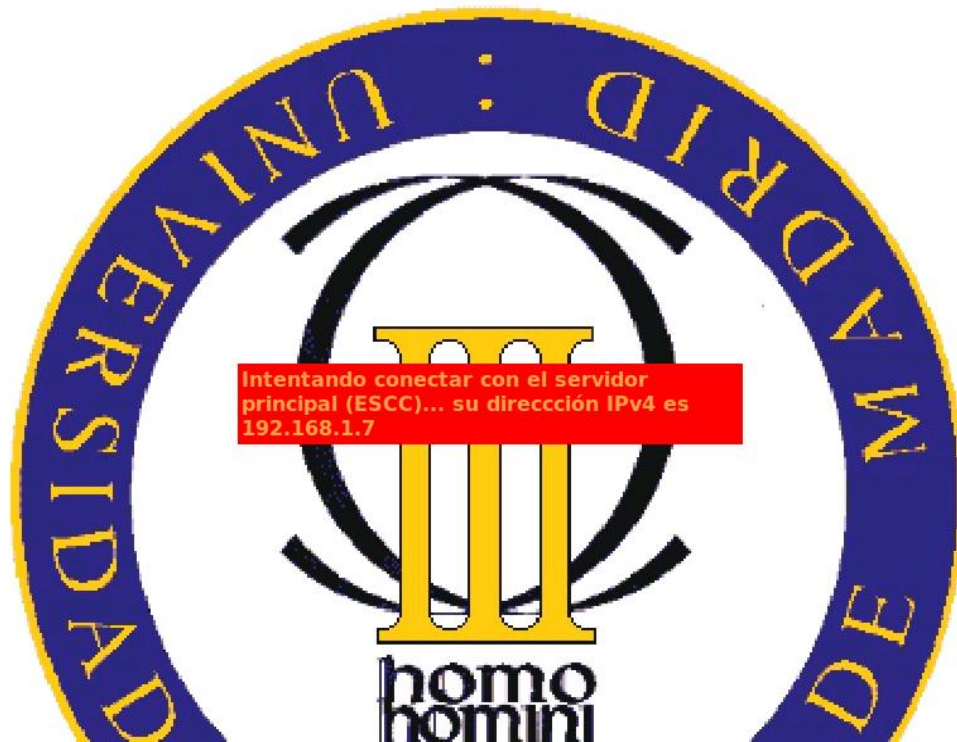


Ilustración 46.-. Presentación generada en la pantalla del *ERC*, en ausencia de conexión con el *ESCC*, al iniciar el equipo.

Esta página (*pantallaInicio.php*) está continuamente refrescándose y se almacena de forma local en el *ERC*.

```
<meta HTTP-EQUIV="REFRESH" CONTENT="5;URL=./pantallaInicio.php"/>
```

Dentro del *script php* que genera el código *html* que presenta la página, se intenta una conexión a la *base de datos* del *ESCC*.

- Si el resultado es negativo no se hace nada y se vuelve a intentar pasados 5 segundos.
- Si por el contrario se obtiene conexión, se redirige la petición sobre el *script repEscPHP.php* del *ESCC*, cuya dirección *IPv4* (*\$IP_serv*) está almacenada en la *base de datos* según las configuraciones realizada por el administrador del sistema mediante la *Interfaz Gráfica de Usuario* web de configuración del *ERC*.

Por tanto, este *script* (*pantallaInicio.php*) está continuamente (cada 5 seg.) monitorizando si el servidor puede proporcionar la configuración del *Escenario* de la *Ubicación*.

Se ha programado que esa comprobación corresponda al análisis de la conectividad con la *base de de datos* del servidor configurado como *ESCC*. En esta misma sección de código *php* se pueden realizar otra serie de comprobaciones, como que el *servidor socket* está activo o que es posible la actualización del contenido multimedia o incluso un conjunto de estas comprobaciones; sin embargo, se ha decidido que con verificar la conectividad con la *base de datos* del *ESCC* se puede considerar que dicho equipo está activo.

La página a la cual se llama, al obtener la conectividad, es la siguiente:

```
window.location.href='http://$IP_serv/div-dat/div-  
local/repEscPHP/repEscPHP.php?IP_serv=$IP_serv&IP_backup=$IP_backup'
```

De este modo si se obtiene un resultado positivo, se realiza una petición *http* sobre un *script php* (*repEscPHP.php*) en el *ESCC*. Este *script* devuelve código *html* con aplicaciones *swf* embebidas, que se ejecutan mediante el plugin *Flash Player*, y que muestra el contenido multimedia de acuerdo al *Escenario* configurado de la *Ubicación*, la cual depende de la *dirección IPv4* del *ERC* que realiza la petición. El esquema lógico de funcionamiento de la página devuelta por el *ESCC*, cuando se obtiene la conectividad, se muestra en el siguiente sub-apartado.

5.4.4.4.3 Presentación de contenido Multimedia en los ERC

Una vez se ha considerado que se tiene conectividad con el *ESCC*, el *ERC* redirige las peticiones hacia el *ESCC* (*repEscPHP.php*) como se ha mostrado en el sub-apartado anterior. De modo que se establece, a partir de este punto, una *arquitectura cliente-servidor* ya que el código *html* generado por el *script repEscPHP.php*, almacenado en el *ESCC*, se descarga y ejecuta en el navegador del *ERC*.

La lógica y el formato del *script php* (*repEscPHP.php*) que genera la página que contiene el *Escenario* configurado de la *Ubicación*, se muestran (de forma esquemática) en la Ilustración 47 y el código (*html*) que la define se recibe directamente desde el *ESCC* mediante el protocolo *http*:

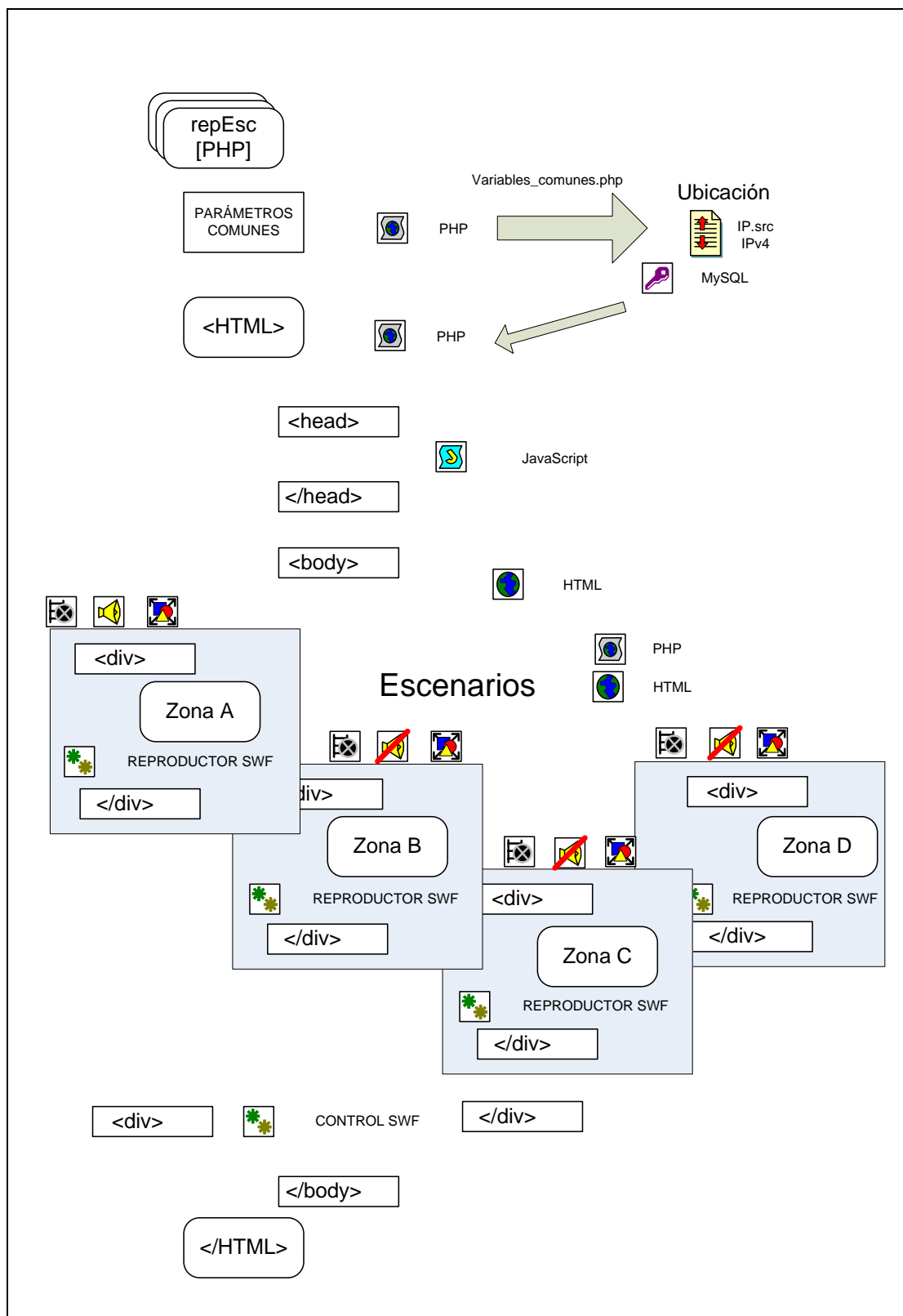


Ilustración 47.-. Resumen del script que genera la página de presentación de contenido multimedia (`repEscPHP.php`).

Cuando el servidor (ESCC) recibe la petición, el script `repEscPHP.php` realiza una serie de comprobaciones para identificar las características de la página que debe devolver y, a continuación, envía una página web que permite mostrar el contenido multimedia a través de

las diferentes *Zonas* del *Escenario* de la *Ubicación* para cada *ERC*. El resultado se genera de acuerdo a la configuración establecida por el administrador del sistema (ver ANEXO C.I.1 MENÚ UBICACIONES del manual de configuración del ESCC). Las comprobaciones se realizan a través de los parámetros recibidos (*GET* o *POST*), además del origen de la petición (*dirección IPv4* origen). El origen de la petición es el parámetro configurado que permite diferenciar el contenido, ya que en la configuración del ESCC se asocia la *dirección IPv4* a la *Ubicación* a la que pertenece el *ERC*.

Para determinar la composición de la página web a mostrar, se ha de identificar la *Ubicación*. Esta información es la más relevante ya que de ella dependen las proporciones de los `<div>`²⁵ y el contenido que se proyectará en cada *Zona*. Este dato (*Ubicación*) se obtiene a través del cálculo de lo que se ha denominado *variables comunes* en el servidor. Así, cuando se accede a una página del servidor que incluye el archivo *variable_comunes.php*, se obtiene la *Ubicación* para la cual se debe proporcionar la configuración:

- Si se está realizando una previsualización, se envía siempre la *Ubicación* como parámetro *GET* en la petición por lo que no hace falta calcularla.
- Si se accede desde las páginas de configuración del ESCC, también se envía la *Ubicación* seleccionada y si no se envía, estamos tratando la configuración por 'defecto'.
- Si la petición se realiza desde un *ERC*, la *Ubicación* se calcula a partir de la *dirección IPv4* desde el equipo²⁶ desde donde se realiza la petición *http*.

Como requisito funcional de diseño se ha establecido que se superponga unas zonas a otras estableciendo la mayor prioridad de visualización a la *Zona A* (zona principal) y por último la *Zona D*. Por tanto, si se quiere programar un fondo dinámico, una forma de hacerlo es configurar la *Zona D* a pantalla completa y programar en esta zona los archivos de fondo que se quieren representar. También se ha establecido que la *zona* que mantiene el sonido activo sea la *Zona A* de la configuración. Todas las *Zonas* contienen la misma *aplicación REPRODUCTOR SWF* (ver Ilustración 47) que se ejecuta en el *plugin Flash Player* (*repZona.swf*). A la aplicación *REPRODUCTOR SWF* (*repZona.swf*) se le pasa, desde el *script php* (*repEecPHP.php*), los parámetros de configuración *Ubicación*, *Escenario* y *Zona* sobre la que tiene que realizar las peticiones de configuración al ESCC.

Finalmente, se introduce mediante un nuevo `<div>`, una aplicación denominada *CONTROL SWF* (ver Ilustración 47), cuya función es monitorizar el estado del *socket* para así tomar decisiones en función de la información leída (*controlSocket.swf*) desde el ESCC. Las dimensiones de esta división son del 0% (tanto de alto, como de ancho), por lo que esta aplicación será invisible para el público real que se encuentre delante del *SPC*.

5.4.4.4.3.1 Aplicación Reproducción de Zona en los ERCs

La aplicación *REPRODUCTOR SWF* (*repZona.swf*) que se muestra en la Ilustración 47 es la encargada de reproducir y mostrar el contenido multimedia. Para lograrlo mediante el *Marco* diseñado, la *aplicación swf* para la *Reproducción de Zona* (*REPRODUCTOR SWF*) se

²⁵ `<div>` Es un *tag HTML* que permite a los autores asignar ciertos atributos a bloques de contenido. En el proyecto fin de carrera es utilizado para incluir cada *Zona* del *Escenario* de acuerdo a unas dimensiones porcentuales almacenadas en la *base de datos* a accesibles a través de la *Interfaz Gráfica de Usuario*.

²⁶ Debido a que se utiliza la *dirección IPv4* origen de la petición, no se permite diferenciar los equipos tras la configuración de un router con *NAT*. La solución a este problema se mantiene como línea de trabajo futuro.

ejecuta en el navegador de forma embebida e independiente, haciendo uso del *plugin Flash Player* sobre el navegador *Mozilla Firefox*, por cada una de las *zonas* presentes en la configuración de la *Ubicación* a la que pertenece el *ERC*. Así, si la *Ubicación* tiene cuatros zonas (máximo permitido) se realizarán peticiones de la configuración desde cada una de ellas, así como petición del contenido de forma independiente.

En el siguiente esquema se resume el funcionamiento de la aplicación:

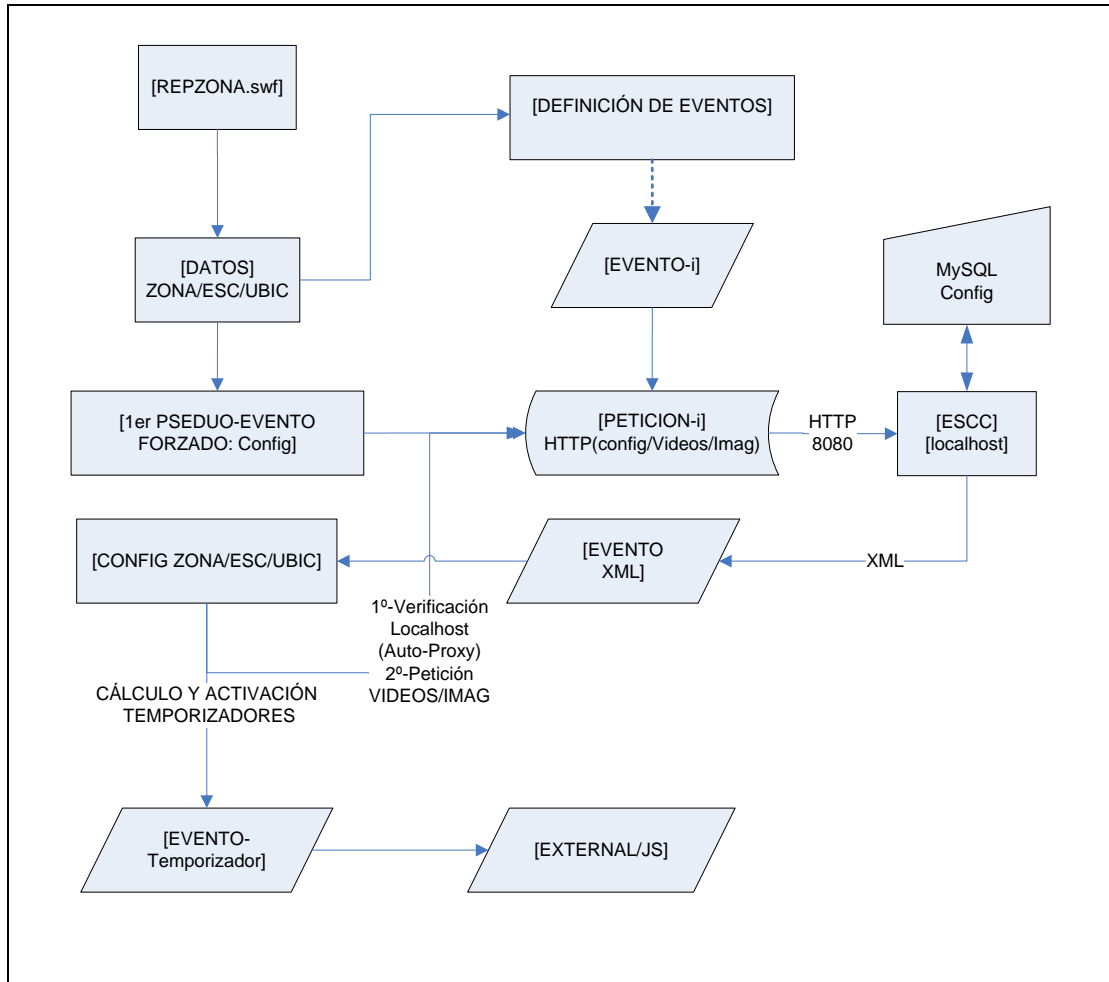


Ilustración 48.-. Resumen de la Aplicación REPRODUCTOR SWF. La aplicación permite reproducir el contenido multimedia de acuerdo a la configuración almacenada en el ESCC.

La aplicación *repZona.swf* (*REPRODUCTOR SWF*) recibe los datos de la *Zona*, el *Escenario*, *Ubicación* y *dirección IPv4* del *ESCC*, al ser pasados éstos como parámetros a la propia aplicación por el script *repEscPHP.php* que presenta el *ERC* a pantalla completa. Estos parámetros, que se introducen en el código *html* generado, son utilizados para formar de modo apropiado las diferentes peticiones realizadas al *ESCC* desde la aplicación *REPRODUCTOR SWF* y así obtener la configuración de los archivos que se deben reproducir.

De este modo, mediante una primera petición *http* al servidor y la obtención de la respuesta *XML* (en la que se obtienen las listas de reproducción actual junto con el nombre de los archivos a reproducir y los parámetros), el *Reproductor de Zona* obtiene los datos necesarios para comenzar a establecer la reproducción. Así, una vez se han recuperado el listado de los archivos a reproducir, la aplicación realiza dos acciones que se establecen en paralelo:

- Comprueba si los diferentes archivos del intervalo a reproducir los tiene almacenados de forma local (*http/XML*) marcando la disponibilidad de éstos. Posteriormente, según la marca, realizará las peticiones al *ESCC* o *localhost* en el momento que los tenga que reproducir vía *HTTP Streaming*.
- Calcula los dos temporizadores, uno para volver a realizar una petición al *ESCC*, para obtener la nueva lista de reproducción, y el otro para calcular cuando debe recalcular los archivos a reproducir en el siguiente intervalo configurado de la lista de reproducción actual.

Una vez la aplicación tiene almacenado el listado de archivos del intervalo actual a reproducir y marcado si lo tiene almacenado de forma local, comienza a realizar las peticiones *HTTP Streaming* para la reproducción del contenido en la *Zona*. La reproducción de las imágenes se realiza en función de los parámetros configurados en la *Interfaz Gráfica de Usuario* y, en la funcionalidad actual, se ha marcado el tiempo que se debe mostrar la imagen en la *lista de reproducción*.

Existen otras muchas funcionalidades y mejoras que pueden añadirse al reproductor, como es la posibilidad de reproducir otros formatos como *swf* o la posibilidad de establecer texto en movimiento a modo de marquesina; sin embargo, son funcionalidades adicionales en el *Marco* y se presentan como líneas de trabajo futuro.

5.4.4.4.3.2 Aplicación Control swf y Clientes Socket en los ERCs

La aplicación *CONTROL SWF* (*controlSocket.swf*), que se muestra incluida en la Ilustración 47, permite controlar en los *ERCs* las órdenes escritas desde el *ESCC*. Estas órdenes se van escribiendo en función de la configuración establecida en el *ESCC* desde un equipo remoto (*EACCC*).

A continuación (Ilustración 49), se presenta un esquema simplificado del funcionamiento de la aplicación de *CONTROL SWF*:

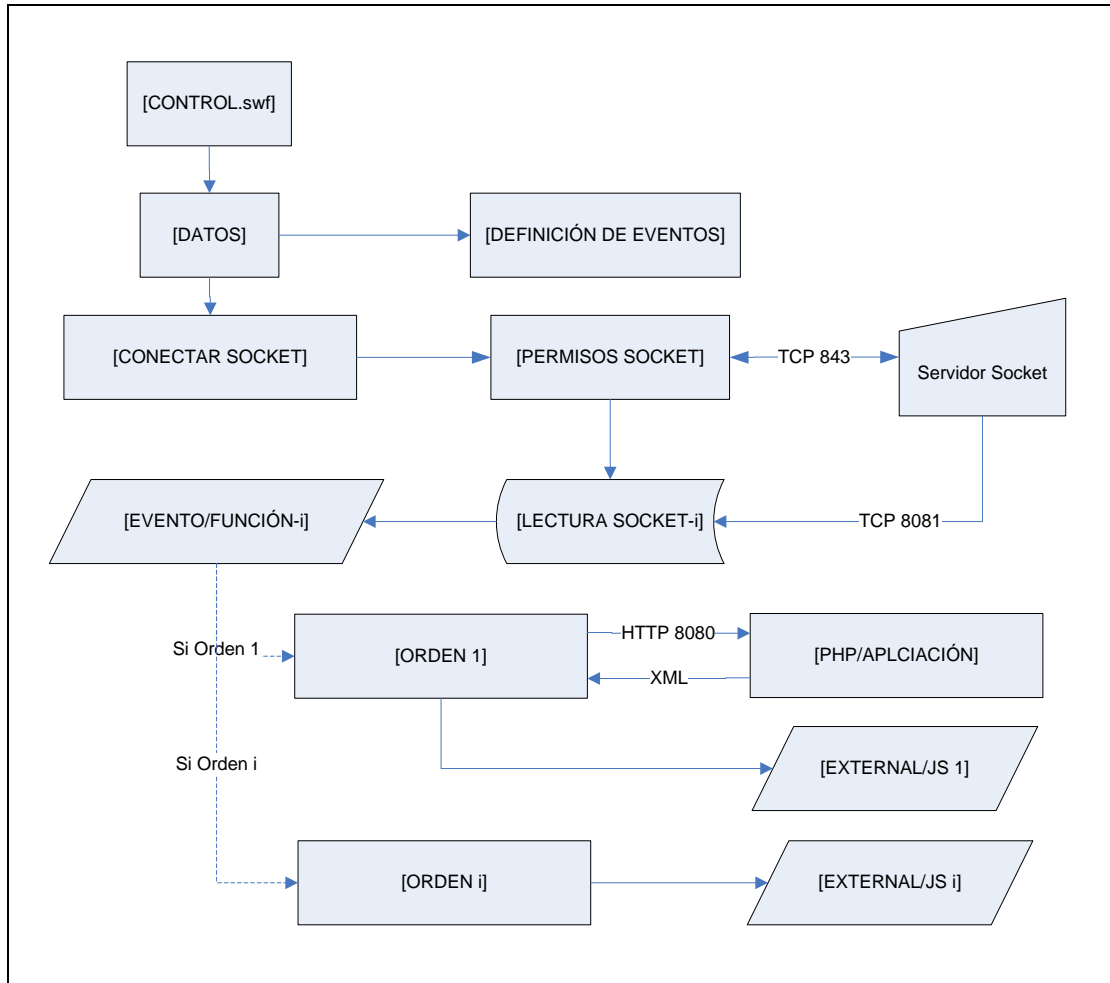


Ilustración 49.-. Resumen de la Aplicación CONTROL SWF. El Cliente Socket permite recibir las órdenes generadas desde el ESCC las cuales se interpretan en la aplicación en el ERC.

Básicamente, el funcionamiento de la aplicación de *CONTROL SWF* consiste en registrar una serie de eventos que pueden ocurrir y elegir, en base a ellos y un procesamiento del evento ocurrido, a qué funciones *JavaScript* se le proporciona el control de la ejecución a través de la clase *ExternalInterface* de *ActionScript 3* (ver Ilustración 49).

Según la evolución del *Marco*, si se incluyeran nuevas funcionalidades, podrían definirse y programarse otros muchos eventos y casuísticas asociadas a dichos eventos; incluso relacionados con potenciales aplicaciones de *Difusión Selectiva Activa* o *Interactiva*. Por ejemplo, sensores controlados por una *aplicación local* que, al detectar determinada información, escribiera en un *socket* también local del *ERC*. De esta forma, al estar registrada una casuística en el evento asociado se podría llamar a una función *JavaScript* que realizara determinada acción relacionada con la información del sensor. Se mencionan algunas de estas posibles implementaciones en el apartado Conclusiones y Trabajos Futuros.

En la Ilustración 49 se muestra el flujo normal de la aplicación. Según va detectando las órdenes en el *socket* se establecen determinadas acciones. Algunas con la interacción de las aplicaciones a través de peticiones *localhost* (*http/XML*) y otras simplemente utilizando la clase *ExternalInterface* para proporcionar el control a las funciones *JavaScript* definidas en *repEscPHP.php*.

Por tanto, en el *Marco* desarrollado se ha considerado en la aplicación *CONTROL SWF* la siguiente secuencia general:

- El *ERC* identifica que debe realizar una nueva petición porque haya detectado (a través del *servidor socket*) que pertenece a otra nueva *Ubicación*, que la *Ubicación* ha sido modificada o bien que deba sincronizar el contenido multimedia de forma local.
- Por tanto, al recibir información del *socket*, se produce un evento registrado en la *aplicación swf* que está controlando el estado del *socket*.
- Mediante la clase *ExternalInterface*, en la *aplicación swf*, se programa la llamada a las distintas funciones *JavaScript* en función del evento que se haya producido en la *aplicación swf*.

Esta aplicación *swf* (*CONTROL SWF*) que se introduce de forma oculta²⁷, realiza una serie de comprobaciones en el *socket* al que se conecta para, así, tomar decisiones de actualización y presentación. Estas decisiones pueden ser asociadas a dos tipos:

1. Peticiones al *ESCC* de nueva configuración (recarga de la página)
2. Peticiones de actualización del contenido multimedia, porque se haya detectado que se ha añadido nuevo o se haya eliminado algún archivo en el *ESCC*

La *aplicación CONTROL SWF* se rige por eventos y, básicamente, realiza las siguientes comprobaciones:

- Evento pérdida de conexión con el socket: la aplicación *swf* comprueba cada 3 segundos el estado de la conexión al *servidor socket*.
- Evento recepción de datos en el socket:
 - *Orden de Ubicación*: El *ERC* comprueba si el equipo pertenece a la *Ubicación*. Si es así se evalúa la orden, si no, se ignora. Actualmente existe una sola orden y es la de actualizar la petición al *ESCC* para recuperar la configuración.
 - *Orden a la dirección IPv4*: El *ERC* verifica si la orden va asignada a su propia *dirección IPv4*. Si es una orden dirigida a él se evalúa y si no, se ignora. Actualmente existe una sola orden y es la de actualizar la petición al *ESCC* para recuperar la configuración. Esta orden se establece, normalmente, cuando el equipo ha variado de forma importante su configuración: por ejemplo, se asigna el equipo a una nueva *Ubicación*.
 - *Orden de Sincronización de Contenido*: El servidor ha enviado una orden indicando que se ha de actualizar el contenido multimedia debido a que, o bien se ha cargado un nuevo archivo en el servidor o se ha eliminado de éste. Posiblemente, cuando se elimina un archivo, éste podría estar asociado a una o varias *listas de reproducción* y, por tanto, a varias *ubicaciones*, por lo que también se escribe en el *socket* una *Orden de Ubicación* que se interpreta en el *ERC*.

²⁷ En las tareas de desarrollo de la aplicación *CONTROL SWF* se ha programado para que muestre información de salida para tareas de resolución de problemas. Para ello, se ha dimensionando el `<div>` que lo contiene con valores adecuados (distinto de 0% e alto y ancho) en el script `repEscPHP.php` que general el código *html* que define la pantalla de presentación.

La aplicación de *CONTROL SWF* está continuamente monitorizando el estado del *servidor socket* por lo que se ha programado un cliente que continuamente está verificando la conectividad.

5.4.4.4 Desarrollo Arquitectura Auto-Proxy

Debido a que se ha diseñado y desarrollado un *Marco para Aplicaciones de Difusión Selectiva* con la característica de una *arquitectura funcional Auto-Proxy*, se ha de establecer cierta lógica en los *ERCs* para alcanzarla.

Dentro del *Marco* desarrollado, los *ERCs* hacen una petición *HTTP Streaming* sobre el archivo que se quiere reproducir en cada *Zona* de cada *ERC*. En diversas pruebas realizadas, y dependiendo de la calidad de la grabación de cada contenido, se ha comprobado que el sistema puede quedar saturado debido a que peticiones simultáneas pueden hacer que el *ancho de banda* de la *WAN* (entre el *ESCC* y el *ERC*) sea superado. Se ha diseñado y desarrollado que cada *ERCs* descargue el contenido de forma local y siempre intente reproducirlo (vía *HTTP Streaming*) en un primer intento desde el propio equipo. Así, las características más importantes para obtener la *arquitectura Auto-Proxy*, se establecen en obtener el contenido almacenado de forma local mediante la aplicación *rsync* y realizar las peticiones de forma local mediante la aplicación *REPRODUCTOR SWF*.

A modo de resumen, la secuencia de funcionamiento del sistema ideado es la siguiente:

- El *ESCC* y los *ERCs* mantienen sincronizado el contenido multimedia de forma automática a través de *rsync*. De este modo, el contenido multimedia subido y presente en el *ESCC* es copiado de forma local en cada *ERC*.
- Los *ERCs* realizan una petición al *ESCC*, de la *lista de los archivos a reproducir* desde la *aplicación swf* de cada *Zona*, *REPRODUCTOR SWF*, cargada en el navegador del *ERC* que presenta el contenido. El *ERC*, mediante la *aplicación REPRODUCTOR SWF*, realiza una petición *http/XML* a sí mismo (*localhost*), para comprobar que el archivo está almacenado de forma local en ese *ERC*. El *script php* que recibe la petición es similar a los programados en el *Motor PHP* y devuelve contenido *XML* con un *flags* indicando qué archivos están almacenados de forma local.
- Cuando el *ERC* recibe la lista de archivos a reproducir sabe cuáles de ellos tiene que reproducir de forma remota y cuales puede reproducir de forma local debido al *flag* marcado en la lista *XML* de reproducción de contenido.

La configuración y el desarrollo que permite la *arquitectura funcional Auto-Proxy* establecen un uso más eficiente del *ancho de banda* del sistema en una situación estable. Este mecanismo, adicionalmente, proporciona otras ventajas como es la reproducción del contenido, aún cuando se haya perdido la conexión con el *ESCC* ante una caída de la *WAN*.

5.5 Resumen

Este capítulo ha resumido el diseño y desarrollo del *Marco* para aplicaciones de *Difusión Selectiva*, el cual es el objetivo inicial del proyecto.

El *Marco* desarrollado para obtener la *arquitectura funcional* descrita en el Capítulo 4, se ha diseñado con un doble propósito:

- *Marco Diseñado y Desarrollado para Establecer la Configuración.* Se han configurado y desarrollado varias aplicaciones para proporcionar la configuración de la aplicación de *Difusión Selectiva*.
- *Marco Diseñado y Desarrollado para proporcionar el Funcionamiento (Sincronización y Reproducción).* Se han configurado y desarrollado varias aplicaciones para proporcionar la reproducción del contenido multimedia y alcanzar la *arquitectura funcional Auto-Proxy* definida en el Capítulo 4. Para alcanzar dicha *arquitectura funcional* es necesario que el contenido multimedia esté almacenado de forma local en el *ERC*, por lo que se ha diseñado en el *Marco* un mecanismo de sincronización y recuperación del contenido en el *ERC*.

En el diseño del *Marco* se ha evitado restringir el desarrollo y las futuras modificaciones a una tecnología concreta. Para ello, se ha abordado el desarrollo mediante una estrategia similar a la utilizada en la elaboración de protocolos de comunicaciones, aunque a nivel de aplicaciones independientes: cada *capa* (a nivel de aplicación) se comunica con las adyacentes de acuerdo a un orden pre-establecido.

Se ha concluido, en el estudio sobre las diferentes alternativas, que las tecnologías más convenientes y que se han utilizado en el desarrollo del *Marco* son:

- *Virtualización (VMWare):* Nos permite simular varios equipos sobre el mismo *sistema operativo* anfitrión. Estos equipos virtualizados pueden ejecutar cualquier sistema operativo que soporte la arquitectura *hardware* virtualizada. El uso de esta herramienta nos permite establecer un entorno de desarrollo y de pruebas más flexible.
- *Ubuntu Server (ESCC) y Ubuntu Desktop 10.10 (ERC):* Estos son los *sistemas operativos* sobre los que se ha desarrollado el *Marco*. La elección de sistemas operativos basados en *Linux* es muy apropiada debido a que es un sistema robusto, estable, rápido y está muy bien documentado. Además, los sistemas basados en *Linux* disponen de una gran cantidad aplicaciones de libre distribución que pueden ser utilizadas en el desarrollo de un *Marco*.
- *Flash Player y ActionScript 3:* Ha sido la tecnología principal elegida para desarrollar la *interfaz gráfica del usuario* administrador en el *ESCC*, la tecnología utilizada para establecer la comunicación con el *servidor socket* y la reproducción del contenido multimedia en el *ERC*.
- *PHP.* Lenguaje en el lado del servidor que permite generar código *html* de forma dinámica, permite la interacción desde las *aplicaciones swf* con el *modelo de datos* a modo de *gateway* y permite la interacción con las aplicaciones complementarias que se ejecutan sobre los equipos principales (*ESCC* y *ERC*) para, así, alcanzar el funcionamiento de la *arquitectura funcional*.
- *MySQL.* Sustentará el modelo de datos de forma persistente de las aplicaciones que se ejecutan sobre el *Marco* diseñado. Permite *modelos de datos* relacionales y dispone de herramientas como *phpMyAdmin* que facilitan su gestión.
- *Mozilla Firefox:* Se ha escogido un navegador para realizar la presentación del contenido multimedia a pantalla completa en el *ERC*, de acuerdo a una configuración establecida en el *ESCC*. *Mozilla Firefox* ha resultado ideal debido al número de complementos que pueden ser utilizados y desarrollados sobre el navegador.
- *Complementos (plugin) para Mozilla Firefox:* El *plugin* más importante utilizado en el *Marco* es el *plugin Flash Player* de *Adobe* que permite la ejecución de *aplicaciones swf* sobre el navegador. Estas *aplicaciones swf* se han desarrollado explícitamente para cubrir las necesidades del *Marco* diseñado. La posibilidad de

incluir otros nuevos *plugins* que proveen nuevas funcionalidades y permitan ejecutar terceras aplicaciones es muy importante dentro del *Marco* como se muestra en Capítulo 6.

En este capítulo se han detallado las configuraciones en cada uno de los equipos principales (*ESCC* y *ERC*) que se han definido en la *arquitectura funcional Auto-Proxy*. También hemos explicado la lógica general de las aplicaciones desarrolladas, que junto con la configuración de las aplicaciones de terceros nos ha permitido desarrollar un *Marco* para *Aplicaciones de Difusión Selectiva*.

Se ha hecho hincapié en algunos aspectos relevantes, como es el *servidor socket* implementado en el *ESCC* y que provee el mecanismo de comunicación con los *ERCs* en relación a la programación y difusión del contenido multimedia. Debido a la tecnología utilizada para la configuración de la aplicación de *Difusión Selectiva* sobre el *Marco* y la presentación de contenido multimedia mediante *aplicaciones swf*, tiene relativa importancia el *servidor de archivos de políticas de socket*, el cual publica los permisos para las conexiones al *servidor socket* desde las aplicaciones *swf*. El hecho de incluir cambios importantes de unas versiones a otras en los *plugins* de *Adobe Flash Player* (como sucedió en la evolución del *plugin* reproductor *Flash Player* a la versión 9.0.r124.0) en aspectos tales como la modificación de la *políticas de seguridad* [106][103], puede hacer que el sistema deje de funcionar a nivel de configuración debido a que el *plugin* es el que ejecuta las *aplicaciones swf* desarrolladas. No obstante, para la reproducción de las aplicaciones en el *ERC*, estos problemas se solventarían sin más que mantener el *plugin* sin actualizar para ejecutar las aplicaciones.

Capítulo 6 Modificación del Marco para la Emisión/Recepción de Streaming Media

En este capítulo se plantea una mejora del *Marco* para aplicaciones de *difusión selectiva* diseñado en el Capítulo 5. El objetivo de esta mejora es el de establecer la posibilidad de recibir contenido multimedia mediante *Streaming Media*; abriendo la posibilidad, de ejecutar sobre el *Marco*, otras aplicaciones de *difusión selectiva* diferentes a la *Cartelería Digital* y que requieren esta capacidad. La solución para desarrollar esta mejora no se ha integrado completamente en la implementación del *Marco*, sin embargo en este capítulo vamos a describir el diseño de dicha solución.

Tal y como se ha definido el *Marco* en el Capítulo 5 se han utilizado *aplicaciones swf* (sobre *Flash Player*) para la reproducción del contenido multimedia para la aplicación de *Cartelería Digital*. Esta tecnología ofrece dos alternativas de protocolos de aplicación (ver Capítulo 2) para la recuperación y reproducción de dicho contenido:

- *Streaming Media* sólo se permite el protocolo *RTMP* y sus variantes.
- *Pseudo Streaming* con el protocolo *HTTP*

Por tanto, para alcanzar el objetivo de proporcionar *Streaming Media* en el *Marco* desarrollado, una primera posibilidad podría haber sido la de incluir un servidor *Streaming Media* (*Darwing Streaming Server* [63], *Helix Universal Server* [33], *Wowza Media Server* [32], *Flash Media Server* [31],...) y modificar la aplicación *REPRODUCTOR SWF*(*repZona.swf*), así como la *interfaz gráfica de usuario* para dotar al sistema de la nueva capacidad (ver Capítulo 5). Enfocar la solución de este modo hubiera obligado a establecer un conjunto de modificaciones elevado en el *Marco* y restringir la emisión del contenido a un número reducido de protocolos (*RTMP*) y codificación de audio y video acotada (ver Tabla 3 del Capítulo 2), por estas razones, esta opción fue finalmente descartada. Además, la posibilidad de emitir mediante técnicas *Multicast* puede ser muy conveniente cuando se realizan emisiones en directo en diferentes sedes y no es posible almacenar previamente el contenido en equipos a modo de *proxy*.

En su lugar, y aprovechando que el *Marco* está implementado mediante aplicaciones embebidas sobre el navegador, se ha optado por seguir una estrategia diferente para aumentar así las funcionalidades del *Marco* usando aplicaciones existentes: el uso de *plugins* para *Mozilla Firefox*.

6.1 Plugin VLC para Mozilla Firefox

El complemento de *VLC (VideoLAN Client)* para *Mozilla* [111] resulta ideal para implementar la mejora de reproducir contenido multimedia mediante técnicas de *Streaming Media* en el *Marco* que se ha desarrollado. Es un complemento a modo de *plugin* que se puede instalar en el navegador y que permite muchas de las funcionalidades que proporciona este reproductor, incluyendo la reproducción de flujo *Multicast*.

La solución de *Streaming* de *VideoLAN*, permite una *arquitectura funcional* en la que se tienen diferentes elementos:

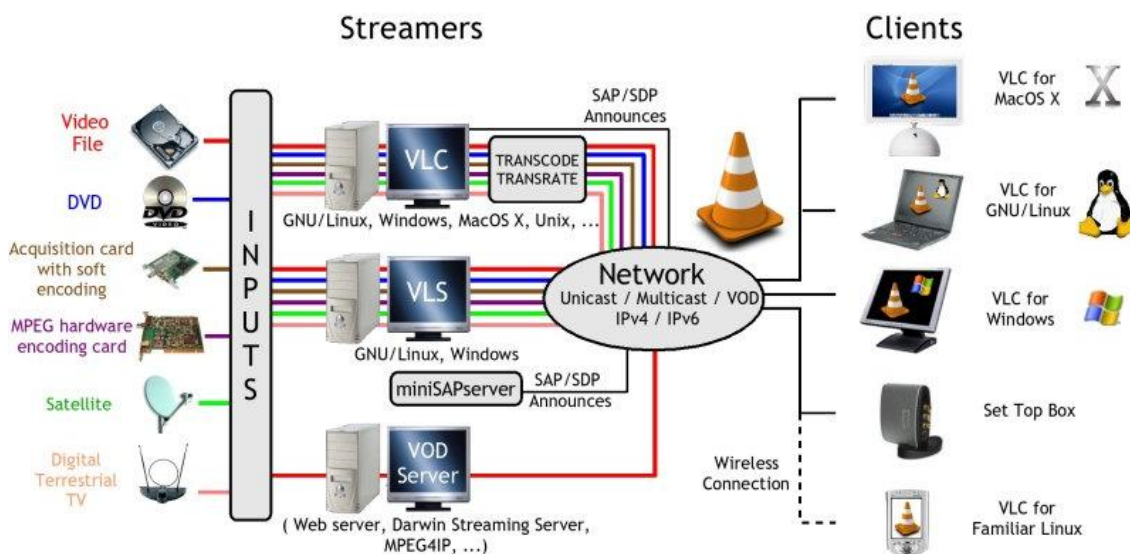


Ilustración 50.- Solución de Streaming propuesta por VideoLAN. Arquitectura Funcional.

Mediante la aplicación *VideoLAN Server (VLS)*, incluida también junto con la aplicación *VideoLAN Client (VLC)*, es posible, entre otros, emitir mediante *Multicast* y *RTP* un flujo *Streaming Media* sobre una red (ver apéndice B.IV Configuración de la aplicación VLC como Emisor Multicast). En la Ilustración 50 también se muestran (como posibles emisores), otros servidores además del VLC o la aplicación VLS, por ejemplo algunos de los *Servidores Streaming* ya mencionados [63] [33] [32] [31] para proporcionar vídeo bajo demanda (*VoD*) y para proporcionar, también, flujos *Streaming Media*.

Mediante el navegador *firefox*, el *plugin* de VLC (complemento de VLC para *Mozilla*) y haciendo un uso adecuado del *API JavaScript* (proporcionado por *VideoLAN* [111]), es posible mostrar contenido multimedia recuperado de un flujo *Multicast*²⁸. El *plugin* de VLC para *Mozilla* exporta varias clases que pueden ser utilizadas para establecer y obtener información. Así, al crearse un objeto del tipo *VLC*, dentro de una página web que presenta el contenido (ver 5.4.4.4.3 Presentación de contenido Multimedia en los ERC del Capítulo 5), se puede hacer uso del *plugin* instalado en el navegador mediante funciones *JavaScript*. De este modo se puede acceder a diferentes funciones definidas en el *API* para el manejo de vídeos, en las que se incluyen reproducción de flujos *Multicast*. Así, los equipos *terminales clientes* que están recibiendo el flujo, pueden recibir y reproducir el contenido multimedia, incluso en directo, si la emisión se establece a través de una tarjeta codificadora de vídeo o directamente de otro elemento hardware (*Satélite, TDT,...*), tal y como se muestra en la Ilustración 50.

6.2 Modificación del Marco con Streaming Media Multicast

En este apartado se presenta la implementación de la solución sobre la modificación del *Marco* desarrollado (ver Capítulo 5). Se presentan tanto las modificaciones que afectan a la *arquitectura funcional* (ver Capítulo 4) como las modificaciones que afectan a la parte del *Marco* para proporcionar la Configuración (ver apdo. 5.2.1 del Capítulo 5) y a la parte del *Marco* que proporciona el Funcionamiento (ver apdo. 5.2.2 del Capítulo 5).

²⁸ En las pruebas realizadas se ha evaluado la mejora sobre flujos *Multicast*; sin embargo, la aplicación permite realizar emisiones *Unicast* mediante varios protocolos.

6.2.1 Modificación de la Arquitectura Funcional

Para proporcionar la recepción de un flujo *Streaming Media* en el *Marco* desarrollado, proporcionando la capacidad de emitir *Multicast*, se ha de introducir nuevos elementos en la *arquitectura funcional*. Estos elementos serían los siguientes:

- Instalar la aplicación *VLC* en el *ESCC* para proporcionarle la capacidad de emitir contenido *Streaming Media*.
- Instalar routers con soporte *Multicast* y configurar algún protocolo, como *IGMP* (*Internet Group Management Protocol* [112] [50] [51]) y/o *PIM* [113] [114], para propagar el flujo entre los routers y subredes y, así, realizar la emisión directamente desde el *ESCC* hacia el resto de sedes donde se encuentran los *ERCs*.

No obstante, para simplificar las pruebas, vamos a utilizar una *arquitectura funcional simplificada* con el objetivo de probar la viabilidad de la propuesta. En esta arquitectura de pruebas añadimos un nuevo elemento, en la red local del *ERC*, que introduce directamente el flujo *Multicast* a todos los equipos presentes en la *VLAN*. A este nuevo elemento lo hemos denominado *Equipo Servidor VLC (ESVLC)* y será el encargado de proporcionar el flujo *Streaming Media Multicast*.

A continuación (Ilustración 51), se muestra un esquema de la *arquitectura funcional* simplificada en la que se introduce el nuevo equipo servidor (ESVLC) para la evaluación de la mejora:

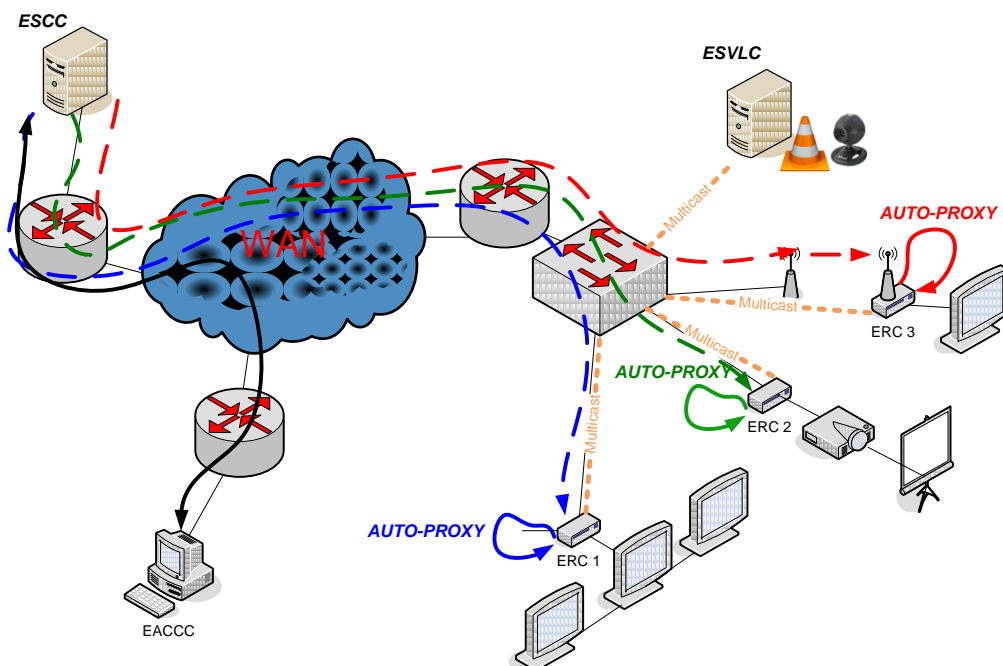


Ilustración 51.-. Arquitectura funcional simplificada con nuevo equipo servidor para proporcionar flujos Multicast en la Red.

Como se puede apreciar en la Ilustración 51 se introduce un nuevo flujo de datos *Multicast* en la red local. Así, sería posible introducir más de un flujo, incluso introducir un flujo con varios canales según el protocolo utilizado, y configurar los *ERC* para que presenten determinados flujos existentes en la red. Como se ha comentado, introducir este nuevo elemento en la red no implica transformar el *Marco* o cambiar de tecnología en la red. Esta

arquitectura simplifica el entorno de pruebas (no es necesario configurar *Multicast* en los *routers*) y no afecta a nuestra evaluación de la funcionalidad proporcionada.

Teniendo en cuenta estos aspectos, la Ilustración 51 muestra la *arquitectura funcional* establecida para evaluar la capacidad del *Marco* de recibir *Streaming Media* a través de *Multicast*, como veremos en las pruebas realizadas (apdo. 9.2 del Capítulo 9, pág. 243) en el entorno de pruebas (apdo. 8.2.5 del Capítulo 8, pág. 213).

6.2.2 Modificación del Marco para Proporcionar Configuración

Además de los cambios considerados en la *arquitectura funcional* se deberían realizar algunas modificaciones en los elementos implicados (*Cliente SEMILIGERO*, *Motor PHP*, *Aplicaciones de Sistema y Modelo de Datos*) en la parte del *Marco para Establecer la Configuración* (ver apdo. 5.2.1 del Capítulo 5, pág. 108). La implementación de la solución ideal debería permitir al usuario a través de una *interfaz gráfica* una serie de configuraciones:

- Configurar de forma remota los parámetros necesarios (*grupo Multicast*, *protocolo de transporte*, *codecs*) para proporcionar emisión *Streaming Media Multicast* desde el *ESCC*.
- Programar la reproducción de flujos *Multicast* sobre las *Zonas*, de los *escenarios* de cada *Ubicación*, de igual modo que se programan los archivos en el *Marco* desarrollado.

Sin embargo, la implementación establecida es limitada y tiene el objetivo de probar la viabilidad de la solución propuesta. Estas limitaciones son:

- La configuración de la emisión se establece de forma independiente y manual con la configuración de la aplicación *VLC* (ver ANEXO B.IV Configuración de la aplicación *VLC* como Emisor *Multicast*). Sin embargo, esta *interfaz gráfica* en el *ESCC* es innecesaria en las pruebas de viabilidad de la propuesta, debido a que la emisión se realiza desde el *ESVLC* (ver 6.2.1 Modificación de la Arquitectura Funcional).
- Ausencia de una *interfaz gráfica* de administrador que le permita configurar la reproducción de los flujos *Multicast* en diferentes *zonas* de los distintos *escenarios* de cada *Ubicación*.

Debido a la ausencia del desarrollo de una *interfaz gráfica*, se ha creado, como veremos (apdo. 6.2.3.2 Configuraciones y Aplicaciones en lado de los *ERCs* para reproducir *Multicast*), una variable para las pruebas que suple esta carencia y que configura el *Marco* para que los equipos *ERCs* puedan recuperar contenido *Multicast* (y por lo tanto hace que el *Marco* funcione en la configuración de pruebas de *Streaming Media Multicast*). La variable contendrá la *dirección IPv4 Multicast*, el *puerto* y el *protocolo* para establecer la comunicación. Si esta variable (*php*) estuviera vacía el funcionamiento sería el explicado en el resto del documento: aplicación de *Cartelería Digital*. Por otra parte, el flujo *Multicast* con presencia de la variable se reproducirá siempre sobre la *zona* principal (*Zona A*, ver apdo. 5.4.4.4.3 del Capítulo 5, pág. 172).

La modificación de la parte del *Marco para Establecer la Configuración* proporciona una configuración fija, ya que se envía el contenido *Streaming Media* desde el servidor *Multicast* (*ESVLC*) a un grupo fijo (*dirección IPv4 Multicast* y *puerto*), se recibe ese grupo en los *ERCs* y se presenta en una *zona* fija (siempre en la *Zona A*) para su reproducción.

6.2.3 Modificación del Marco para Proporcionar el Funcionamiento

Para proporcionar las herramientas para reproducir los flujos *Multicast* se han añadido algunos complementos al *Marco*. Estos complementos son básicamente proporcionar las configuraciones y aplicaciones que permitan al *Marco* la utilización del *plugin VLC* sobre el navegador *Mozilla Firefox* en sustitución del *REPRODUCTOR SWF de Zona* (ver apdo 5.4.4.4.3.1 Aplicación Reproducción de Zona en los ERCs) utilizado en la aplicación de *Cartelería Digital*

6.2.3.1 Configuraciones y Aplicaciones en lado del ESCC para proveer Multicast

Como se ha explicado en el Capítulo 5 (apdo. 5.4.4.4.3 Presentación de contenido Multimedia en los ERC), el script *repEscPHP.php* genera la página *html* que presenta el *Escenario* (con las *zonas*) de la *Ubicación* configurada para cada *ERC*. El código *html* publica elementos embebidos (*aplicaciones swf* que se ejecutan en el navegador del *ERC* mediante el *plugin Flash Player*) y que proporcionan la reproducción de contenido multimedia en la *zona* (*repZona.swf*) y el control de las órdenes sobre el *socket* (*controlSocket.swf*).

Para la evaluación de la solución propuesta se incluye la variable fija (*dirección IPV4 Multicast, puerto y protocolo*) en el script *repEscPHP.php*. Si se ha definido la variable, en vez de incluir en la '*Zona A*' la aplicación *repZona.swf* y su reproducción sobre *Flash Player*, se utiliza el *plugin VLC* para *Mozilla Firefox* y se reproduce el recurso almacenado en la variable fija, en nuestro caso un flujo *Streaming Media Multicast*.

A continuación (Ilustración 52), se presenta un esquema de esta modificación en el *ESCC*; es decir, la modificación del script *repEscPHP.php* (ver Ilustración 47 del Capítulo 5, pág. 173) que proporciona la página *html* cargada en el *ERC* para reproducir *Streaming Media Multicast*:

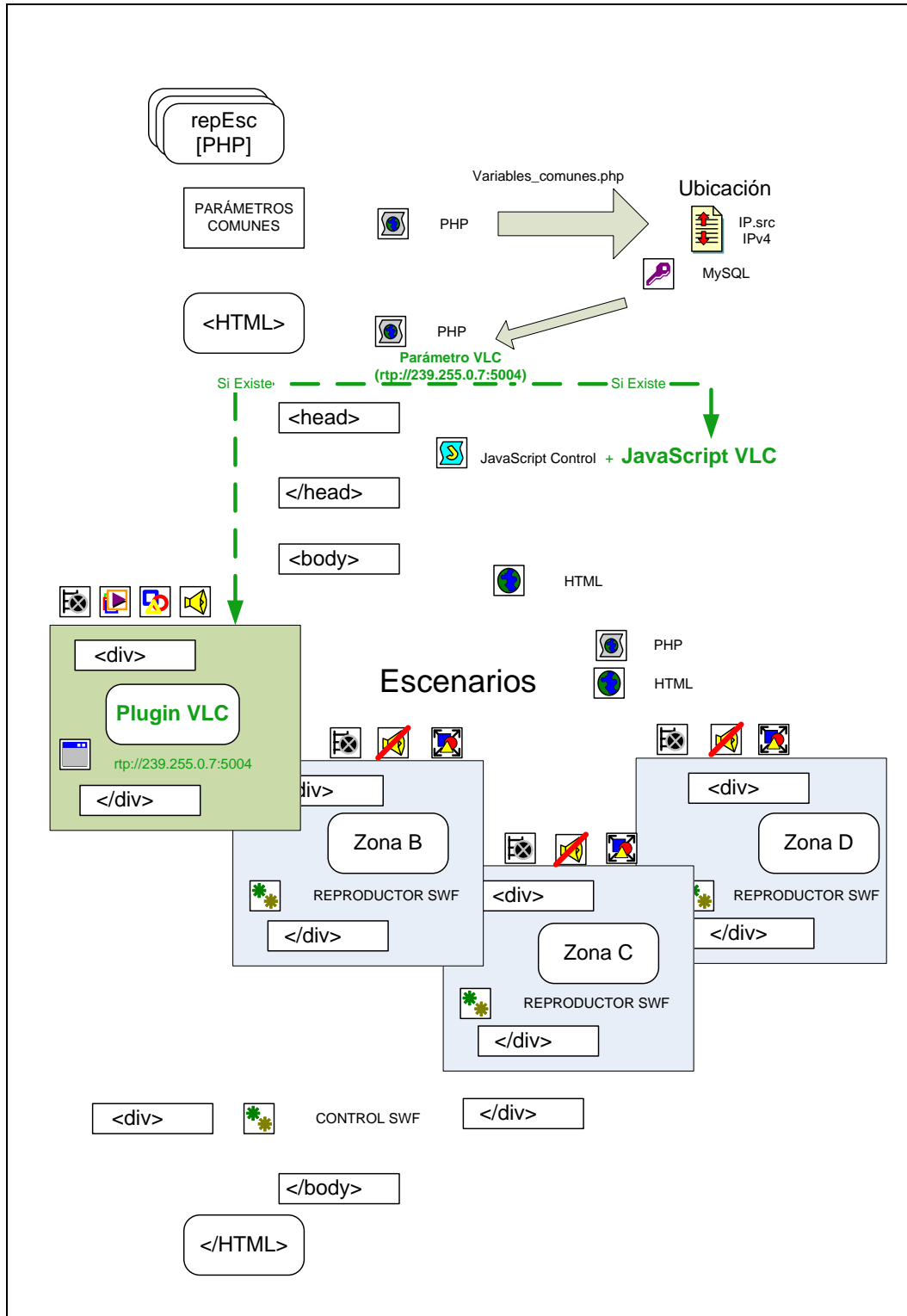


Ilustración 52.-. Resumen del script que genera la página de presentación de contenido multimedia Modificada para introducir y hacer uso del plugin VLC (repEscPHP.php).

El código introducido en el *Marco* permite la reproducción de flujo *Multicast*. Para añadir las funcionalidades del reproductor VLC se deben añadir unas pocas líneas al *script php* que maneja las peticiones de los ERC (*repEscPHP.php*).

Como se ha comentado, según si se ha definido la variable fija, se ejecuta la aplicación desarrollada en el *Marco REPRODUCTOR SWF* (*repZona.swf*, ver apdo.

5.4.4.3.1 Aplicación Reproducción de Zona en los ERCs) mediante el *plugin Flash Player* o se declara un elemento como *VLC (vlc1)* para su uso:

```
$MRL_VLC="";
$MRL_VLC="rtp://239.255.0.7:5004";

[...]

if ($A_ancho !=0 && $A_alto != 0){

echo"
<div id='zona-A' style='position: absolute; z-index: 4;left:$A_x%; top:
$A_y%; width:$A_ancho%; height: $A_alto%; visibility: visible; overflow:
visible;background:#000000'>";

if ($MRL_VLC!="") {

echo"
<embed                                     type='application/x-vlc-plugin'
pluginspage='http://www.videolan.org'
version='VideoLAN.VLCPlugin.2'
width='100%'
height='100%'
id='vlc1'>
</embed>

<script language='javascript'>
    play('vlc1','$MRL_VLC');
</script>
";

}else{

echo"
<embed                                     type='application/x-shockwave-flash'
pluginspage='http://www.macromedia.com/go/getflashplayer'
src='$reproductor?escPHP=$esc&zonaPHP=A&servidorPHP=$servidor&ubi
cacionPHP=$ubicacion' quality='high'
scale='noscale' salign='lt' wmode='transparent'
width=100%
height=100%
name='repZona' align='center' />
";

}

echo"</div >";

}
```

Se ha de indicar que el código adicional introducido con respecto al *Marco* inicial es el que se muestra en negrita. El código declara un elemento (`id='vlc1'`) como tipo *plugin VLC* ('application/x-vlc-plugin') y, seguidamente, se llama a una función *JavaScript* (`play`), previamente declarada, que inicia la reproducción.

La función *JavaScript* definida (`play`) crean un objeto, declarado como elemento de tipo *plugin VLC*, y que permite el manejo de la reproducción haciendo uso de las clases exportadas por el *plugin VLC*:

```
<script language="javascript">
    function play (tgt,uri) {

        var tgt = document.getElementById(tgt);
        tgt.playlist.add(uri,uri, "");
        tgt.playlist.play();
```



```
}
</script>
```

La función *JavaScript* simplemente indica al objeto *tgt*, instancia del *plugin* VLC, que comience la reproducción del flujo almacenado en la variable *uri* (<protocolo>://<dirección IPV4>:<puerto>), cuyo valor corresponderá con el almacenado en la variable *\$MRL_VLC*.

6.2.3.2 Configuraciones y Aplicaciones en lado de los ERCs para reproducir Multicast

El *Marco* presentado en el Capítulo 5 se ha explicado cómo se establece la presentación de contenido multimedia haciendo uso del navegador *Mozilla Firefox* a pantalla completa y realizando la reproducción del contenido mediante *aplicaciones swf* que se ejecutan con el *plugin* *Flash Player*.

El sistema operativo *Ubuntu* viene por defecto con la instalación de *Mozilla Firefox* con un *plugin* que permite la reproducción de contenido multimedia cuando se publica el contenido como *application/x-vlc-plugin*:

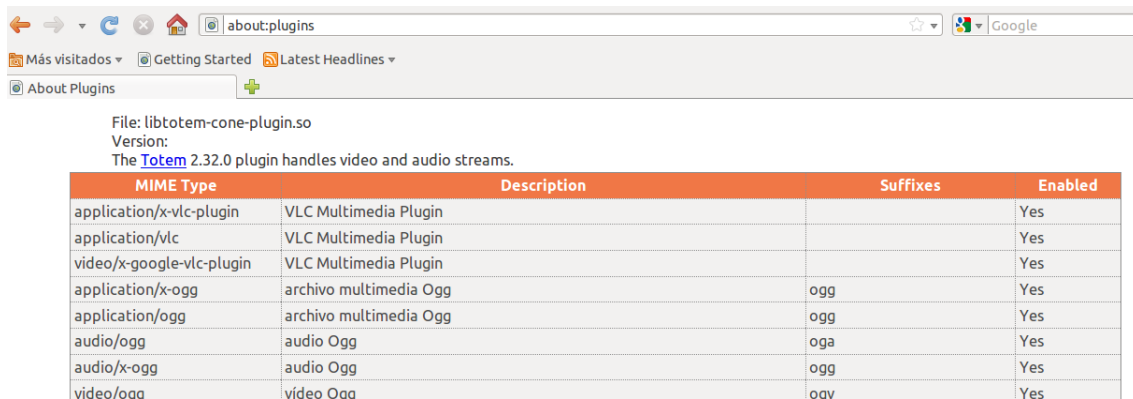


Ilustración 53.-. Plugin por defecto en Ubuntu Desktop 10.10 en Mozilla Firefox.

Sin embargo, el *plugin* por defecto (*Totem 2.32.0*) no es capaz de reproducir flujos *Multicast* presentes en la red, por lo que se desactiva y se instala el *plugin* oficial de VLC. Por ello, en el *ERC*, se hace necesaria la instalación del *plugin* VLC bajo el sistema *Linux*, tal y como se muestra en la documentación de *VideoLAN*. Como se ha comentado ya, en el desarrollo del *Marco*, se ha optado por la distribución *Desktop* de *Ubuntu 10.10*, la cual es una distribución *GNU/Linux* basada en *Debian GNU/Linux*. Así, se siguen los pasos para la instalación de la aplicación sobre *Debian*.

Por tanto, siguiendo los pasos que se indican en el manual de VLC [115], se instala el *plugin* sobre el navegador:

```
pfc@escc:~$ sudo apt-get update
pfc@escc:~$ sudo apt-get install mozilla-plugin-vlc
```

Para verificar su instalación, lo hacemos a través del propio navegador introduciendo el comando apropiado (*about:plugins*) en la barra de exploración:

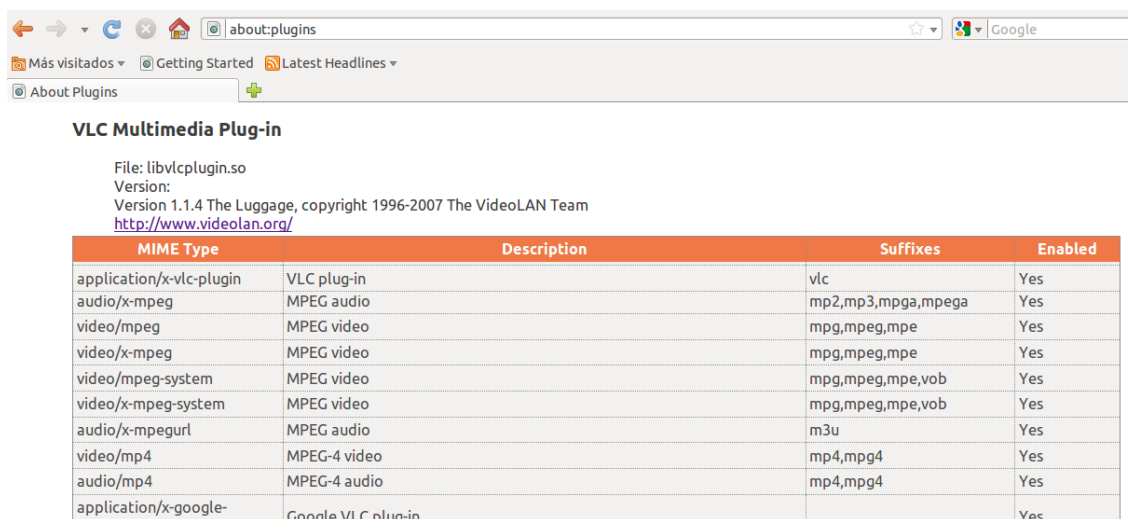


Ilustración 54.-. Plugin VLC 1.1.4 en Ubuntu Desktop 10.10 para Mozilla Firefox.

Así, cuando se etiqueta un objeto en la página *html* con tipo *MIME* como *application/x-vlc-plugin*, será el *plugin VLC* el encargado de establecer la reproducción.

Se ha de tener en cuenta que la instalación del *plugin VLC* para *Mozilla*, también es necesaria en el equipo *EACCC* si se quiere permitir la previsualización de dicho contenido. No obstante, se ha de entender que, debido a la naturaleza de configuración del *EACCC*, desde una sede remota y ajena a donde se encuentran los equipos *ERCs*, es posible que éste no tenga acceso al flujo *Streaming Media* que se está emitiendo en las sedes donde se encuentran los *ERCs*.

6.3 Otros Plugins para Mozilla

A medida que evolucionan los servicios pueden aparecer nuevas necesidades tecnológicas basadas en estándares que permitan ampliar y enriquecer la experiencia del usuario en aplicaciones, por ejemplo de *Cartelería Digital*, o simplemente establecer nuevas funcionalidades. La misma estrategia de uso del *plugin VLC* del navegador para la incorporación de contenido *Streaming Media* al *Marco*, la podemos aplicar a otros *plugins* para obtener otras funcionalidades en el *Marco*.

En este subapartado se identifican otros *plugins* que podrían ser utilizados en el *Marco*.

Ambulant

SMIL [116] es un estándar a tener en cuenta, si se hace uso de tecnologías web para presentar el contenido multimedia, como es el caso del *Marco* desarrollado en el presente trabajo. *SMIL* es el acrónimo de *Synchronized Multimedia Integration Language* (*Lenguaje de Integración Multimedia Sincronizada*) y es un estándar del *W3C* (*World Wide Web Consortium*) para presentaciones multimedia. El lenguaje *SMIL* permite integrar audio, video, imágenes, texto o cualquier otro contenido multimedia [117]. Así, se ha encontrado un reproductor de *software libre*, denominado *Ambulant* [118], que puede ser añadido como *plugin* a *Mozilla* para poder así reproducir el contenido de forma sincronizada de acuerdo al estándar *SMIL 3.0*.

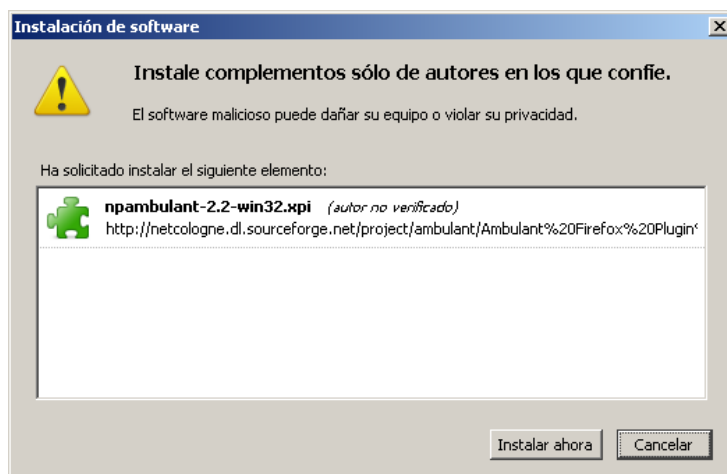


Ilustración 55.-. Plugin para Mozilla Firefox del Reproductor Ambient para SMIL.

QuickTime

QuickTime [119] es un reproductor desarrollado por *Apple* que permite la reproducción de muchos *contenedores multimedia* de videos y audio y que permite, incluso, la reproducción de *Streaming Media* sobre muchos protocolos de transporte.

El reproductor está disponible en forma de *plugin* para *Mozilla* [120] de modo que puede reproducir videos y otro tipo de archivos, entre ellos *SMIL* [121], de forma embebida y de igual modo que con el *plugin VLC*. *QuickTime* dispone de soporte para las funciones *JavaScript* [122] que permiten controlar la reproducción del contenido ya sea de forma manual o añadiendo funcionalidades a la aplicación web. Una característica a considerar de la tecnología *QuickTime* es que han desarrollado un importante trabajo en los reproductores, para permitir reproducir contenido en directo directamente utilizando el protocolo *HTTP*.

Se ha de considerar que el *plugin QuickTime* para *Firefox* no está disponible para el sistema operativo *Linux*, por lo que se ha descartado su utilización en el ámbito del *proyecto fin de carrera*, aunque existe un *plugin* similar denominado *gecko-mediaplayer* [123] que permite también la reproducción de contenido multimedia.

Silverlight

Silverlight [83] es una plataforma similar a *Adobe Flash*, para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad. La plataforma se encuentra disponible para varios navegadores web, entre ellos *Mozilla*, como complemento en forma de *plugin* [124]. Debido a que la tecnología ha sido desarrollada por *Microsoft*, no se ha establecido un soporte real para otras plataformas distintas de *Windows*, aunque sí se ha desarrollado una solución alternativa denominada *Moonlight* [125] y se ha establecido un convenio con *Novell* para su desarrollo [126].

6.4 Resumen

En este capítulo hemos presentado una estrategia alternativa a la que en un principio podría considerarse como natural para proveer al *Marco* la capacidad de reproducir contenido *Streaming Media*.

La estrategia utilizada se fundamenta en la configuración y utilización del *plugin VLC* para *Mozilla Firefox* e incluirlo en el navegador que presenta el contenido multimedia a pantalla completa en el *Marco* presentado en el Capítulo 5.

Las modificaciones generales que se han propuesto sobre la *arquitectura funcional* y el *Marco* permiten, incluso, la reproducción de contenido multimedia estableciendo técnicas *Multicast* para el transporte del contenido y realizando toda la configuración desde una aplicación centralizada en el *ESCC*.

Con el objetivo de evaluar la viabilidad de la estrategia en el *Marco*, reproducir contenido multimedia *Streaming Media Multicast* incluyendo la configuración del *plugin VLC*, sólo se han de realizar algunas modificaciones. En este capítulo se muestra una *arquitectura funcional simplificada* y un *Marco* con funcionalidades reducidas que nos permiten realizar una serie de pruebas para evaluar la idoneidad de modificar el *Marco* con la estrategia presentada.

Hemos evidenciado que, utilizando la estrategia de incluir el *plugin VLC*, se pueden añadir otras funcionalidades si se hace uso de otro tipo de *plugins* disponibles para *Mozilla Firefox*.

PARTE IV. Resultados Obtenidos y Pruebas Realizadas

Capítulo 7 Comparativa: Arquitectura Funcional Implementada y Arquitectura Funcional de Referencia

No existe un sistema de *Difusión Selectiva* perfecto que se adapte a todas las situaciones. Es posible que un determinado sistema, con menores funcionalidades desarrolladas, sea más adecuado para un determinado cliente, que una solución compleja en la que intervienen una gran cantidad de equipos. Sin embargo, en el desarrollo del *Marco* se han tenido en cuenta las diferentes *arquitecturas funcionales* encontradas con la intención de poder adaptarlo a una situación más compleja en donde se introduzcan nuevas aplicaciones y/o servicios.

En este capítulo se pretende establecer una comparativa *de la arquitectura funcional implementada* con la *arquitectura funcional de referencia* estudiada y las aplicaciones que ofrece ésta.

En la comparativa se mostrarán las similitudes y diferencias, tanto en las aplicaciones que se pueden ofrecer, como en el rol de los equipos que se presentan en ambas arquitecturas. También se mostrarán las diferencias y similitudes encontradas en los métodos de funcionamiento y trabajo de ambas arquitecturas, sobre todo cuando se encuentre presente una red WAN entre la sede principal y el resto de sedes.

7.1 Administración de los Equipos

Una característica común a la mayoría de los equipos y sistemas estudiados es que la forma de configurar los diferentes equipos se ha implementado a partir de herramientas web mediante el desarrollo de *interfaces gráficas de usuario*. Así, la configuración de las aplicaciones que se ejecutan, tanto sobre la *arquitectura funcional de referencia* como en la *arquitectura funcional implementada*, se realiza de forma remota. Esto permite una cómoda configuración a partir de, prácticamente, cualquier equipo que disponga de un navegador web.

En la *arquitectura funcional de referencia* se han considerado dos figuras que acceden al sistema. Por una parte el *autor creador de contenido* (*Content Author*), el cual es el encargado de proporcionar el contenido multimedia y la programación de la difusión de contenido y, por otra parte, el *administrador de red* (*Network Management*) el cual sería el encargado de la configuración de todos los equipos de la arquitectura con especial atención sobre los que intervienen en el transporte del contenido multimedia. Dentro del *proyecto fin de carrera*, estas dos figuras no quedan diferenciadas, aunque se han definido usuarios con diferentes privilegios. Se han diferenciado en la arquitectura funcional dos usuarios; uno de ellos con todos los privilegios de acceso (*sistema*) y otro con acceso solamente a la configuración de la aplicación de *Cartelería Digital* (*admin*), tanto en el *ESCC*, como en los *ERCs*. El objetivo de definir dos usuarios con distintos privilegios de acceso es el de mostrar la posibilidad de establecer un sistema multi-privilegios mediante el *Marco* desarrollado, capacidad ésta que podría ser utilizada en el futuro para desarrollar nuevas aplicaciones como veremos.

7.2 Aplicaciones de Difusión Selectiva

La *arquitectura de referencia* ha sido escogida principalmente porque soporta la mayoría de las aplicaciones y modos de trabajo de las diferentes alternativas de *arquitecturas de*

Difusión Selectiva presentes en el mercado. Es decir, la *arquitectura de referencia* no se centra exclusivamente en mostrar videos e imágenes en forma de *Cartelería Digital*, sino que abarca otras aplicaciones. Para ello provee un sistema que permite ofrecer, bajo la misma arquitectura y el mismo modo de funcionamiento, otros servicios similares e interesantes que podrán ir desarrollándose dentro de *arquitectura funcional implementada*. Estas nuevas aplicaciones y/o servicios que se pueden incorporar a la *arquitectura funcional implementada* se deberán basar en el mismo *Marco* con que se ha implementado la arquitectura para *aplicaciones de Difusión Selectiva de Cartelería Digital* en el proyecto fin de carrera. Se ha mostrado, en el Capítulo 6, cómo el *Marco* desarrollado permite incluir nuevas funcionalidades que proporcionan nuevas aplicaciones de *Difusión Selectiva*.

Estas diferentes aplicaciones de una arquitectura de *Difusión Selectiva*, distintas a la *Cartelería Digital*, son: el denominado *Portal Web de Video*, el cual posibilita la emisión de videos seleccionados en función del usuario que accede al portal; y el *Canal de TV Corporativo* que permite la emisión, recepción y reproducción de flujos de *Streaming Media* a través de redes *IPv4* a modo de *IPTV* [6]. Incluso, como veremos, es posible proveer, mediante el mismo *Marco*, aplicaciones de *Difusión Selectiva Interactiva* (ver apdo. 10.2.3 Implementación Difusión Selectiva Interactiva), las cuales en principio no son ofrecidas por la *arquitectura funcional de referencia* de forma genérica.

A continuación, se exponen algunos argumentos que apoyan que, el *Marco* desarrollado, encaja de forma apropiada, para proveer el resto de *aplicaciones de Difusión Selectiva* que ofrece la *arquitectura funcional de referencia*:

7.2.1 *Portal Web de Vídeo*

En la *arquitectura funcional de referencia* se ha desarrollado la aplicación *Portal Web de Video* con una gran cantidad de funcionalidades, desde publicar eventos en directo para dar una clase magistral, hasta la posibilidad de grabar ese contenido publicado para poder reproducirlo en diferido en posteriores audiciones.

El acceso al contenido en el sistema de referencia está basado en perfiles de usuarios. La emisión de un *Portal Web de Video* es de contenido multimedia; es decir, que no son sólo video, sino que se pueden reproducir en muchas ocasiones animaciones en otros formatos. En la *arquitectura funcional de referencia*, una de las características más interesantes es que pueden emitir contenido en directo como parte del contenido publicado. Esto es posible a partir de *Servidores Streaming* que se configuran y se sincronizan con la planificación de las *listas de reproducción* del servidor, donde se establece la configuración general (apdo. 3.4.2.2 Portal (Web) de Video). De esta forma se establece un sistema de *Difusión Selectiva* muy sofisticado.

En la *arquitectura funcional implementada* se ha establecido que el *contenido multimedia presentado* (el cual se presenta mediante una página web a pantalla completa, como se ha explicado en el apartado 5.4.4.4.3 Presentación de contenido Multimedia en los ERC, pág. 172) se proporcione según la *dirección IPv4* desde la cual se realiza la petición y la configuración establecida para ese parámetro; es decir, en función de un parámetro lógico y/o geográfico diferenciador.

Para desarrollar un *Portal Web de Vídeo*, partiendo del *Marco* y de la *arquitectura funcional implementada*, en primer lugar se deberá seleccionar un nuevo²⁹ *parámetro diferenciador* más apropiado a la aplicación en cuestión. Posteriormente, se debería programar en un servidor, el cual sería el *ESCC*, una página web de configuración (en la misma línea de las desarrolladas), en la que el administrador del sistema pueda asociar el *parámetro diferenciador* escogido, con el contenido multimedia a visualizar. Se podría seleccionar, como *parámetro diferenciador*, el *grupo* al que pueda pertenecer el *usuario* que se conecta a la aplicación, y asociar el *grupo* al contenido multimedia que se permite visualizar. Incluso cabría la posibilidad de asociar varios *grupos* a un mismo *usuario*. Esta asociación se podría realizar a través de un interfaz web del mismo modo que se ha realizado para la aplicación de *Cartelería Digital* y así, según sea el *usuario* que accede al sistema, se le permitirá acceder a unos determinados vídeos e imágenes. De esta forma, el *Marco* apenas se vería afectado ya que la aplicación se regiría por el mismo esquema que el desarrollado y no se vería modificado para la aplicación de *Cartelería Digital*.

Un ejemplo representativo de la aplicación del *Difusión Selectiva de Portal Web de Vídeo* es la de asociar, a los *estudiantes y profesores* de una universidad, diferentes *asignaturas*. De este modo un estudiante o un profesor (*usuario*), sólo podría acceder a los vídeos de las asignaturas (*grupos*) que le haya asignado un administrador del sistema (asignatura que imparta un profesor o asignatura en las que esté matriculado un estudiante). Los vídeos podrían ser las clases grabadas de las asignaturas impartidas de forma presencial, material complementario, como ensayo de experimentos, o conferencias relacionadas con la asignatura. Mediante el esquema propuesto en el ejemplo, el propio *usuario profesor*, mediante un perfil de usuario diferenciador, podría administrar el contenido de cada *grupo (asignatura)* al que pertenezca. El mismo esquema es extrapolable a organizaciones empresariales donde los *empleados (usuarios)* son clasificados por *departamentos (grupos)*.

7.2.2 Canal de TV Corporativo

Simplificando la *arquitectura funcional* para presentar la aplicación de un *Canal de TV Corporativo*, no es más que permitir la posibilidad de recibir un flujo *Streaming Media* (normalmente *Multicast*, aunque a veces también como *Unicast* y *Broadcast*) emitido, normalmente en directo, desde un servidor, el *ESCC*, y reproducirlo en los *ERCs*. Estas emisiones deben realizarse obligatoriamente mediante algún sistema *Streaming Media*, debido a que la línea de tiempo de la difusión no debe verse afectada por el *ancho de banda* disponible.

Por tanto, las necesidades básicas de la *arquitectura funcional implementada* para ofrecer un *Canal de TV Corporativo*, son:

- Programar una lista de vídeos o fuentes de video, dentro de las cuales puede haber emisión en directo.
- Emitir los vídeos mediante *Streaming Media (Multicast, Unicast o Broadcast)* desde un servidor.
- Recibir el tráfico *Streaming Media* y reproducirlo en los diferentes *ERCs*.

La *programación de una lista de archivos* se ha realizado, para la aplicación de *Cartelería Digital*, a través de una interfaz web desarrollada en el proyecto fin de carrera. Por lo tanto, el *Marco* desarrollado es apropiado, ya que ha sido utilizado y testeado para este propósito.

²⁹ Se deduce que, para desarrollar una aplicación de un *Portal Web de Video* con funcionalidades de *Difusión Selectiva*, resulta más apropiado establecer como *parámetro diferenciador* otro diferente a la *dirección IPv4*.

Los problemas *a priori* son: emitir los videos programados en las listas a través de flujos *Streaming Media*, ya sea *Multicast* o cualquier otro método, recibir de forma correcta el contenido multimedia y reproducir el flujo mediante los *ERCs*.

Estos escollos, *emisión*, *recepción* y *reproducción*, son fácilmente salvable dentro del *Marco*, precisamente debido a la forma de implementarlo:

- **Emisión:** Se ha mostrado en el Capítulo 6 que aplicaciones muy populares y abiertas, como *VLS (VideoLAN Server)*, permiten actualmente realizar emisiones sobre varios protocolos *Streaming Media* y mediante varios métodos (incluido *Multicast*) sobre redes *IPv4* e *IPv6*. La aplicación *VLS* (presente actualmente en *VLC, VideoLAN Client*) también permite, incluso, detectar tarjetas *DVB-S* o *DVB-T* y emitir su contenido a través de flujos *Streaming Media*. Ésta aplicación también puede ser controlada a través de la línea de comandos, lo que facilita la integración de *VLC* con el *Marco* desarrollado: se ha mostrado, en el Capítulo 5 que es posible invocar aplicaciones desde la *interfaz gráfica* de forma remota. Así, es posible activar la emisión de contenido *Streaming Media* desde el propio *ESCC* mediante la programación de la *interfaz gráfica de usuario* que realice llamadas a la aplicación *VLC*. Por tanto, la estrategia para establecer este control sobre la emisión *Streaming Media* sería la misma que la utilizada para la aplicación de *Cartelería Digital*; es decir, a través del desarrollo de una *interfaz gráfica web*, un usuario administrador a través de un *EACCC* podría realizar llamadas a *scripts php* que se ejecutan en el *ESCC*. Estos *scripts php*, a su vez, pueden ejecutar la aplicación *VLC* de acuerdo a una configuración establecida y recuperada de las tablas de la *base de datos* y establecer así el método (*protocolos, códecs*) para propagar (*Multicast, Unicast*) los flujos *Streaming Media* que los *ERCs* deberán decodificar y reproducir.
- **Recepción:** Para que el *ERC* reciba los flujos de forma correcta mediante técnicas *Multicast* la *arquitectura funcional* se debe apoyar en una configuración apropiada de los elementos de red a través de protocolos tales como *IGMP* o *PIM*. Estas consideraciones se han presentado en el Capítulo 6.
- **Reproducción:** El problema de la reproducción del contenido multimedia a través del flujo *Streaming Media* recibido, también puede ser solventado gracias a la aplicación *VLC* (apdo. 6.1 Plugin *VLC* para *Mozilla*). Como se ha mostrado en el Capítulo 6, los programadores de ésta han desarrollado un *plugin* para el navegador *Mozilla Firefox* que permite el manejo de flujos *Streaming Media* para su reproducción sobre el navegador (ver apdo. 6.2.3.2 Configuraciones y Aplicaciones en lado de los *ERCs* para reproducir *Multicast*). Existen también otras aplicaciones disponibles en la actualidad a modo de *plugin* (*Ambulant, Quicktime*) que son capaces de ser integradas en un navegador web (ver apdo. 6.3 Otros Plugins para *Mozilla*) y pueden ser utilizadas principalmente para la reproducción del contenido.

7.2.3 Difusión Selectiva Interactiva

La funcionalidad adicional de interactividad en aplicaciones en la *arquitectura funcional de referencia* como es el hecho, por ejemplo, de disponer de un mando a distancia en el *DMP* también podría ser añadida a la *arquitectura funcional implementada*, añadiendo nuevas funcionalidades al *Marco* en el *ERC* y modificando ligeramente la *arquitectura funcional*.

Para lograrlo, se debería desarrollar una aplicación en el *ERC* que fuera capaz de recibir e interpretar las instrucciones de un mando a distancia a través de un sistema infrarrojo como

puede ser *LIRC (Linux Infrared Remote Control)* [127] [128]. Según las órdenes recibidas a través del mando a distancia, esta misma aplicación debería conectarse y escribir en un *servidor socket local* del *ERC*, del mismo modo que se hace en el actual sistema de *Difusión Selectiva* para *Cartelería Digital*. Así, en el *ERC* debería ejecutarse un *servidor socket* de forma *local*, al cual se conectaría una *aplicación swf de control local* embebida en el navegador web que pudieran interpretar órdenes y cambiar, según la orden escrita por la aplicación, el canal reproducido del flujo *Multicast* recibido.

La funcionalidad de interactividad mediante mando a distancia ha sido provista en la *arquitectura funcional de referencia*; sin embargo, mediante el *Marco* desarrollado y con la estrategia que se ha comentado en este apartado en el párrafo anterior, es posible incluso abordar otro tipo de interactividad en la aplicación de *Difusión Selectiva*. Así, la aplicación que escribe en el *servidor socket local* del *ERC* puede ser cualquier aplicación que interactúe con elementos *hardware* externos: sensores, cámaras,... y, de este modo, la interactividad podría obtenerse mediante pantallas táctiles, temperatura externa, genero del público objetivo o cualquier otro mecanismo de decisión que se nos pueda ocurrir; siendo mucho más flexible que la interactividad provista en la *arquitectura de referencia*.

Mediante una estrategia como la comentada, y aplicando las mismas ideas y estrategias que las utilizadas para el *Marco* desarrollado, es posible implementar, además de *Difusión Selectiva Pasiva*, también *Difusión Selectiva Adaptativa o Activa* y *Difusión Selectiva Interactiva*, tal y como se han definido en el apartado 2.5 del Capítulo 2 (Definición de Tipos de Difusión Selectiva), lo que convierte al *Marco* en un sistema ideal para ofrecer servicios mucho más ambiciosos que los ofrecidos por la *arquitectura funcional de referencia*.

7.3 Rol de los Equipos en las Arquitecturas Funcionales

Tanto en la *arquitectura funcional de referencia*, como en la *arquitectura funcional implementada* existen equipos cuyo rol es similar. Esta característica es debida a que ambos sistemas implementan una arquitectura *Cliente-Servidor*.

Es por ello que en ambos sistemas aparecen equipos con roles similares:

7.3.1 Equipos con Rol Servidor

El rol del equipo servidor en ambas arquitecturas se asocia tanto para la distribución del los archivos multimedia, así como para la distribución de la configuración.

En la arquitectura funcional de referencia este doble rol puede estar separado en varios equipos (*Digital Media Manager*, *Video Portal Appliance Server*, *Digital Media Encoders*, *Video on Demand*, *Streaming Servers with Storage*; ver apdo. 3.4.3 Arquitectura Funcional de Referencia del Capítulo 3). Estos equipos se encuentran en el llamado *Canal de Administración y Publicación*, para la provisión de la configuración y en los *Elementos de Codificación del Contenido Multimedia*, para la generación y provisión del contenido multimedia.

En la *arquitectura de referencia* se establece la posibilidad de configurar cada *DMP (Digital Media Player)* de forma independiente para que emitan contenido multimedia de acuerdo a una programación de archivos que se almacenan de forma local. Sin embargo, en el sistema desarrollado, la *arquitectura funcional* necesita que exista, en todas las situaciones, un equipo con rol servidor (*ESCC*), el cual aprovisiona la configuración y el contenido al resto de equipos (*ERCs*). Esta característica disponible en la *arquitectura funcional de referencia*, podría añadirse al sistema de difusión selectiva desarrollado, sin más que copiar del *ESCC* los script y

las configuraciones en cada *ERC* de modo que cada equipo funcionaría como un *ESCC/ERC*. Estas pruebas de integración se han considerado como líneas de trabajo futuro.

En el caso de la *arquitectura funcional implementada*, para la aplicación desarrollada de *Cartelería Digital*, es un solo equipo el que proporciona el contenido multimedia y la configuración a los *equipos terminales (ERC)*. Este equipo es el *ESCC (Equipo Servidor de Configuración y Contenido)*. En el Capítulo 6 han introducido nuevos equipos para la generación de contenido cuando se pretende establecer una aplicación en la que se incluyen otros mecanismos de aprovisionar contenido multimedia como es *Streaming Media* mediante, por ejemplo, *Multicast*. Sin embargo, las funcionalidades del equipo denominado *Equipo Servidor VLC (ESVLC)* pueden desarrollarse sobre el propio *ESCC*.

7.3.2 Equipos con Rol Reproductor

El rol del equipo reproductor o *equipo terminal* se establece en ambas arquitecturas en un único equipo

Se han definido varios modelos de *equipos terminales* dentro de la *arquitectura funcional de referencia*. Estos modelos dependen de la aplicación que se está ejecutando en el sistema, por ejemplo, si la aplicación es *Canal de TV Corporativo* el *DMP (Digital Media Player)* puede disponer de un mando a distancia (*DMP 4400G*, *DMP 4310G*), mientras que si se trata de *Cartelería Digital* el *equipo terminal* carece de esta posibilidad (*DMP 4305G*). El resto de diferencias entre los *DMPs* desarrollados en la *arquitectura de referencia* son evoluciones *hardware* y *software* que mejoran las prestaciones; sin embargo, el rol de estos equipos es siempre el mismo: reproducir el contenido multimedia de acuerdo a la configuración establecida en los equipos con rol servidor.

De la misma forma, en la *arquitectura funcional implementada* el rol de *equipo terminal* lo satisface el denominado *ERC (Equipo Reproductor Cliente)*. Su funcionalidad es la de reproducir el contenido multimedia según se ha configurado en el *ESCC* y sólo se ha desarrollado un solo modelado. Se ha obviado la definición de las especificaciones *hardware* para los *ERCs*, ya que se ha realizado su desarrollo bajo máquinas virtuales de forma que su rendimiento varía según en el equipo físico en el que se esté ejecutando. Tan sólo se han definido las necesidades *software* que el equipo necesita para ejercer sus funciones, al contrario que en la *solución de referencia* en el que el *hardware* de estos equipos ha sido explícitamente definido.

En la *arquitectura de referencia*, los equipos reproductores del contenido multimedia, los *DMPs*, en cualquiera de sus versiones, almacenan de forma local parte del contenido multimedia disponible en los equipos con rol servidor. De este modo, mantienen contenido multimedia disponible de forma local para reproducirlo en caso de que se interrumpa la comunicación con los equipos que proporcionan dicho contenido multimedia, estableciendo un sistema de alta disponibilidad y acotado a cierto contenido. Por otra parte, en el caso de la *arquitectura funcional implementada*, los equipos descargan todo el contenido estático disponible en el *ESCC*, por tanto, la disponibilidad es alta de acuerdo a la configuración establecida en caso de perder la conexión de forma momentánea. Se ha de indicar que podría ser necesaria la monitorización constante de la conectividad con el *ESCC*, de modo que podrían ocurrir situaciones conflictivas en la *arquitectura funcional implementada*, por ejemplo, cuando cambia el intervalo de reproducción y no hay disponible conectividad con el *ESCC*. La solución de este problema se plantea como mejora futura en el apartado 10.2 Líneas de Trabajo Futuro.

Otra característica a tener en cuenta es que en la *arquitectura funcional implementada* los *equipos reproductores (ERCs)*, deben mantener una *dirección IPv4* estática y conocida debido a

que será este el *parámetro diferenciador* en la reproducción de contenido; mientras que en los *equipos reproductores* de la *arquitectura funcional de referencia* (DMPs), pueden incluso establecer un funcionamiento adecuado si adquieren esa dirección IP a través del protocolo DHCP (Dynamic Host Configuration Protocol) [129]. En la *arquitectura funcional de referencia* se asocia como parámetro diferenciador selectivo la *dirección MAC* o dirección física del DMP, mientras que el parámetro diferenciador que configura el contenido en la *arquitectura funcional implementada*, está asociado a la *dirección IPv4* o la dirección lógica de ERC.

7.4 Métodos de Funcionamiento y Trabajo

Dentro de la solución de la *arquitectura funcional de referencia* se han encontrado, fundamentalmente, dos *modos de funcionamiento* a los que se han denominado *Espacio de Funcionamiento Simple* y *Espacio de Funcionamiento Distribuido*.

Como se ha comentado en el Capítulo 3, esta distinción es debida a la presencia de una red intermedia que limita el *ancho de banda* disponible (WAN), desde alguno de los DMPs hacia los equipos con *Rol Servidor* y viceversa. En la *arquitectura de referencia* en un *Espacio de Funcionamiento Distribuido*, se han de introducir nuevos equipos en la arquitectura en cada una de las sedes, con el objetivo de reducir el consumo del *ancho de banda* hacia el servidor. Estos equipos que se denominan WAE (Wide Area Application Engine), con diversos matices según el rol (*Root*, *Standalone*, *Edge*) que implementen (ver apdo. 3.4.3 Arquitectura Funcional de Referencia), utilizan protocolos que permiten mejorar el consumo de *ancho de banda*. El protocolo WCCP que capturar las peticiones al servidor de forma local, establece un portal cautivo para las peticiones del protocolo HTTP (control y contenido), que hace que estos nuevos equipos, en parte, funcionen como un *servidor proxy* dentro de la arquitectura. De esta forma, si tras el equipo WAE de la sede, hubiera varios DMPs, el *ancho de banda* consumido se reduce de forma considerable en determinadas aplicaciones, en comparación con que no existiera este equipo. La introducción de estos equipos es necesaria y hace variar el funcionamiento de la arquitectura cuando se introducen en el sistema, ya que las peticiones se controlan desde un equipo de cada sede (WAE).

Por otra parte, en la *arquitectura funcional implementada*, se utiliza una estrategia diferente a incluir nuevos elementos en la arquitectura. Para alcanzar un aprovechamiento óptimo del *ancho de banda*, se modifica la implementación, tanto del ESCC, como de los ERCs, estableciendo una evolución. Así, en vez de establecer un nuevo elemento de red en la sede remota que haga las funciones de *proxy*, se provisiona a los ERCs a que, en una primera instancia, intenten recuperar el contenido multimedia de ellos mismos (*arquitectura funcional Auto-Proxy*). De esta forma, si obtienen éxito envían las peticiones para recuperar el contenido directamente sobre ellos mismos (*localhost*) y en caso contrario lo hacen sobre el ESCC. Esta estrategia finalmente adoptada hace que los ERCs, descarguen todo el contenido multimedia del ESCC; no obstante, solamente realizan la descarga de cada archivo una sola vez por cada equipo de la sede.

Debido a que, en la *arquitectura funcional de referencia*, es necesario introducir nuevos equipos (cuando se introduce una WAN entre sedes), se ha variado la nomenclatura en la *arquitectura funcional implementada* para referirnos a la situación en que existe una WAN que separa el *servidor* de los *equipos reproductores*. En la *arquitectura funcional implementada* denominamos *espacios de trabajo* a las situaciones en las que opera el sistema de *Difusión Selectiva* desarrollado sobre la red; de modo que si existe una WAN entre el ESCC y algún ERC, estamos ante un *Espacio de Trabajo Distribuido* y, en ausencia de ésta, nos encontramos en un *Espacio de Trabajo Simple*. La *arquitectura funcional implementada* actúa del mismo modo en ambos *espacios de trabajo* por lo que no es necesario incluir nuevos equipos. En la

arquitectura funcional de referencia se denominan *espacios de funcionamiento*, en lugar de *espacios de trabajo*, y es necesario incluir en estos espacios diferentes equipos en según qué situación de operación nos encontremos.

Comparando ambas alternativas de forma cualitativa, la *arquitectura de referencia* implementa un sistema similar al que se ha denominado *arquitectura proxy-caché* en el Capítulo 4 (apdo. 4.4.1 Arquitectura Proxy-Caché). Se observa que quizás la solución de referencia obtenga una mayor eficiencia sobre el *ancho de banda WAN*, ya que el contenido multimedia se descarga una vez por cada sede, en vez de por cada *equipo terminal* de cada sede; sin embargo, ese contenido se ha de descargar siempre que se quiera reproducir (desde el equipo WAE hacia los DMPs), lo que equivale a suponer que no es necesario optimizar el *ancho de banda* de la LAN. Además, la *arquitectura funcional implementada* obtiene eficiencia, de forma automática y sin ningún tipo de configuración, aún cuando no exista una WAN intermedia y se esté en un *Espacio de Trabajo Simple*, tal y como se ha definido.

7.5 Resumen

En el capítulo hemos mostrado algunas características que se han considerado relevantes y que son comunes o divergentes en la *arquitectura funcional* implementada, con respecto a la *arquitectura funcional* de la solución de referencia de Cisco.

Hemos obtenido algunas conclusiones que se derivan de éste análisis:

- El método de acceso a la configuración mediante herramientas web es muy acertado debido a la facilidad que se ofrece en el acceso a los administradores del sistema. Este método de acceso mediante estas herramientas común tanto a la *arquitectura funcional de referencia* como a la *arquitectura funcional implementada*.
- En la *arquitectura funcional de referencia* se hace una distinción entre dos roles de usuario que administran los sistemas. Por una parte, el rol de usuario que realiza la creación y la administración de contenido multimedia y, por otra parte, el administrador de la red. Esta distinción suponemos que es debida a que el número de elementos en la *arquitectura de referencia* varía al incluir una WAN (en el camino de comunicación entre los equipos terminales y los equipos con rol servidor) y es el administrador de red el que debiera dar de alta estos nuevos equipos y realizar la configuración para proporcionar, al menos, la conectividad. En el caso de la *arquitectura funcional implementada* se ha obviado esta doble distinción ya que los equipos que afectan al sistema de *Difusión Selectiva* son siempre los mismos. En la aplicación para la administración del sistema desarrollado se ha creado un usuario con todos los privilegios y otro usuario que sólo tiene acceso a los privilegios para configurar la aplicación de *Difusión Selectiva*. Este doble acceso evidencia la posibilidad de crear una aplicación basada en diferentes usuarios utilizando el *Marco* y se ha considerado que también podría resultar útil para la configuración del sistema.
- Las funcionalidades de las aplicaciones de la *solución de referencia* son, además de variadas, más sofisticadas y permite una mayor versatilidad en la configuración. Esta sofisticación es debida a que es una solución comercial cuyo objetivo es proporcionar mayores funcionalidades que su competencia. Por otra parte, en la solución de *Difusión Selectiva* que se ha presentado en este proyecto, tiene como objetivo el de proveer un *Marco* en el que desarrollar aplicaciones para *Difusión Selectiva*. Por tanto, añadir nuevas funcionalidades y aplicaciones sobre el *Marco*

desarrollado queda enmarcado fuera del objetivo de proyecto y pueden plantearse como líneas de trabajo futuro.

- Debido al diseño del *Marco* para implementar la *arquitectura funcional*, es posible proveer nuevas aplicaciones usando las mismas técnicas que las utilizadas para la aplicación de *Cartelería Digital* desarrollada para probar el *Marco*. Además, sobre la *arquitectura funcional* del *Marco* se pueden desarrollar otro tipo de aplicaciones ausentes en la *arquitectura de referencia*, como son las aplicaciones de *Difusión Selectiva* en la que se incluye interactividad más allá de uso de un mando a distancia.
- Debido a que las arquitecturas funcionales de ambas soluciones tiene algunos aspectos en común, como es estar basado en un sistema *cliente-servidor*, se identifican varios equipos con el mismo rol. Sin embargo, la *arquitectura funcional de referencia* provee diferentes funcionalidades en diferentes equipos lo que complica la solución, siendo la *arquitectura funcional implementada* más simple y más fácil de mantener.
- Hemos diferenciado claramente el hecho de que sea necesario introducir nuevos elementos que intervienen en la *arquitectura funcional de referencia* al incluir una *WAN* entre los equipos terminales y servidores. La *arquitectura funcional de implementada* permanece idéntica si se incluye una *WAN*, lo cual nos muestra de nuevo la sencillez, facilidad de mantenimiento y escalabilidad con respecto a la *arquitectura funcional de referencia*.
- Dentro de la solución desarrollada en este proyecto, se ha evidenciado la importancia de asignar una *dirección IPv4* estática a los *ERCs* (*Equipos Reproductores Clientes*) ya que será el parámetro diferenciador en la *Difusión Selectiva*, mientras que en la solución de referencia estudiada puede hacerse mediante técnicas de provisión de *dirección IP* automáticas, ya que el parámetro diferenciador es la *dirección MAC* del *DMP* (*Digital Media Player*). Así, para grandes despliegues la *arquitectura de referencia* toma ventaja con respecto a la implantación de la *arquitectura funcional implementada*, debido a que no es necesario establecer la configuración de cada equipo de forma manual.

Capítulo 8 Entorno de Pruebas

Las pruebas realizadas para evaluar el sistema se han centrado fundamentalmente, además de verificar el funcionamiento correcto, en comprobar si las características introducidas en términos de eficiencia de *ancho de banda* contribuyen realmente a un mejor aprovechamiento de los recursos en el *Marco*. De este modo, las pruebas han estado enfocadas a mostrar las diferencias entre las esperadas mejoras de la arquitectura *Auto-Proxy* y la arquitectura inicial. Así, se han evaluado dos posibles modos de trabajo del *Marco*, el cual es condicionado por la arquitectura de red: *Espacio de Trabajo Simple* y *Espacio de Trabajo Distribuido* (definidos en los apartados. 4.2.3.1 y 4.2.3.2).

Para realizar las pruebas con cierto rigor se ha utilizado tanto *software* que permite virtualizar los equipos de una forma rápida, como *hardware* que nos ayuda a generar una arquitectura de red acorde a una situación real. Para ello, se ha instalado y configurado un entorno de pruebas lo más parecido posible a una típica situación de trabajo empresarial o institucional.

Por último, se han definido diferentes tipos de pruebas para forzar el *Marco* a una situación en la que, éste, funcione como si no se hubiera implementado la arquitectura *Auto-Proxy*, con el objetivo de evaluar la *arquitectura funcional* del *Marco* para la aplicación de *Cartelería Digital* desarrollada.

8.1 Resumen del Entorno de Pruebas

8.1.1 Equipamiento Disponible para Entorno de Pruebas

El entorno de pruebas configurado se ha realizado con el siguiente equipamiento hardware:

Equipamiento	Modelo	Nomenclatura	Descripción
Routers	1 Router Cisco 1721	Router A	El router soporta el protocolo IEEE 802.1Q (a veces referido como dot1q). Se utilizará para encapsular las distintas VLANs en modo trunk con el switch. IOS: c1700-y7-mz.124-15.T13.bin Memoria: RAM/Flash: 64M/32M
	1 Router Cisco 1750	Router B	El router de la sede donde se encuentra el ESCC. No soporta el protocolo dot1q. IOS: c1700-y-mz.121-27b.bin Memoria: RAM/Flash: 16M/8M
Switch	1 Switch Cisco Catalyst 2950	SWITCH	Switch configurado para emular diferentes conmutadores de cada sede. IOS: 2950-i6k2l2q4-mz.121-22.EA12.bin Memoria: RAM/Flash: 16M/8M PID: WS-C2950T-24
AP Inalámbrico	FON2100A/B/C	AP	Firmware: OpenWrt Kamikaze With X-Wrt Extensions 7.09 Kernel: Linux 2.6.21.5 Device: La Fonera (ACCTON MR3201A) Board: Atheros AR2315
Ordenadores Personales	1 Toshiba Tecra A10	Equipo Anfitrión 1	SO: Windows XP SP3
	1 Toshiba Tecra S3	Equipo Anfitrión 2	SO: Windows XP SP3
Cableado	Cables RJ-45	---	Para la interconexión de los distintos equipos.
	1 cable DB60 DTE/DCE	Serial-WAN-Serial	Para la emulación de la red WAN

Tabla 17.-. Equipamiento Utilizado en el Entorno Pruebas

Con el hardware disponible se completa la configuración del entorno de pruebas en cada una de las configuraciones de los espacios de trabajo establecidos. La conexión lógica y física de estos equipos se detalla en el apdo. 8.2 Esquema Lógico/Físico del Entorno de Pruebas.

8.1.2 Aplicaciones Utilizadas en el Entorno de Pruebas

Se han utilizado las siguientes aplicaciones software para generar y evaluar el Marco sobre cada entorno de pruebas:

Aplicación	Versión	Descripción
VMWare Server	v2.0.2	VMWare es un sistema de virtualización por software. Estas aplicaciones de virtualización permiten crear y ejecutar máquinas virtuales sobre sistemas operativos anfitriones. Es decir, proporcionan un ambiente de ejecución similar, a todos los efectos, a un computador físico (excepto en el puro acceso físico al hardware simulado), con CPU (puede ser más de una), BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro (pueden ser más de uno), etc. Tanto VMWare Server como VMWare Player están disponibles de forma gratuita.
VMWare Player	v3.1.4	

Aplicación	Versión	Descripción
VideoLAN Client (VLC)	v1.7 Windows v1.4 Ubuntu	Es una aplicación <i>software</i> libre distribuida bajo la licencia <i>GNU GPL</i> . Soporta varios codecs de audio y video, así como diferentes tipos de <i>contenedores multimedia</i> , además soporta los formatos de <i>DVD</i> , <i>VCD</i> y varios protocolos <i>Streaming</i> . No necesita la descarga e instalación de codecs en el sistema puesto que los lleva incorporados.
Wireshark	v1.2.6 (SVN Rev 31702)	Es una aplicación del tipo analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones. Permite configurar la tarjeta de red en modo promiscuo permitiendo ver todo el tráfico que pasa por una determinada interfaz de red. <i>Wireshark</i> es <i>software</i> libre, y se ejecuta sobre la mayoría de sistemas operativos <i>Unix</i> y compatibles, incluyendo <i>Ubuntu</i> , así como en <i>Microsoft Windows</i> .
iperf	V2.0.5	Es una aplicación ampliamente usada para test de redes de comunicación que permite generar y recibir flujos <i>TCP</i> y <i>UDP</i> . Permite medir varios parámetros de red en función del flujo emitido/recibido en equipos distintos separados por la red a evaluar.
Origin Pro	v8.0724 (v8 SR0)	Es una aplicación que permite analizar y mostrar datos de forma muy versátil. Permite incluir varios ejes sobre la misma gráfica y además realizar operaciones matemáticas sobre los datos mostrados.
Media Info	v0.7.38	Es una aplicación de <i>software</i> libre que permite mostrar datos técnicos de la codificación de archivos multimedia, así como mostrar las etiquetas sobre muchos de los contenedores multimedia disponibles.

Tabla 18.- Aplicaciones Utilizadas en el Entorno de Pruebas.

Las aplicaciones presentadas en Tabla 18 son utilizadas para generar el entorno de pruebas, así como para analizar el comportamiento del *Marco* sobre éste.

8.1.3 Espacios de Trabajo Evaluados en el Entorno de Pruebas

Se han evaluado varias configuraciones para verificar el funcionamiento del sistema, éstas son:

1. **Configuración del Espacio de Trabajo Simple Virtualizado:** Se ha conectado y configurado el *entorno de pruebas* en una situación en la que no está presente una *red de área amplia* (WAN) y se ha supuesto, en esta situación, que estuviesen todos los equipos del sistema concentrados dentro de un solo emplazamiento físico. Esta simulación se ha realizado mediante equipos virtualizados y el único interfaz de red configurado del *Equipo Anfitrión 1* ha sido una interfaz virtual. De este modo, se simula un *switch* que interconecta los equipos, a partir de la aplicación para virtualización (VMWare), sobre una máquina anfitriona con *sistema operativo Microsoft Windows XP*. (Detalle en apdo. 8.2.1 Espacio de Trabajo Simple Virtualizado)
2. **Configuración del Espacio de Trabajo Simple:** Para un análisis más real de la arquitectura, se ha evaluado el sistema sobre equipamiento de red físico utilizando varios equipos anfitriones (*Equipo Anfitrión 1* y *Equipo Anfitrión 2*) para los equipos desarrollados (ERCs y ESCC). Se han realizado las pruebas configurando, dentro del entorno de pruebas, un *espacio de trabajo simple* con encaminamiento LAN hacia dos subredes diferentes. De este modo, el *ancho de banda* del sistema queda limitado básicamente por la electrónica de *nivel 2* (o capa 2) de la ubicación física. El *ancho de banda* en una LAN, típicamente, son 100Mbps; aunque cuando intervienen elementos inalámbricos esta *ancho de banda* será menor. Además, aparecen retardos que introducen los *routers* que se encuentran en el camino entre los dos equipos, y que intervienen en los tiempos a la hora de enviar/recibir datos al igual que en una situación real. (Detalle en apdo. 8.2.2 Espacio de Trabajo Simple).
3. **Configuración del Espacio de Trabajo Distribuido:** Se ha configurado también el *entorno de pruebas* en una situación donde se simulan diferentes emplazamientos físicos separados por una WAN. Con esta configuración se han realizado diversas

pruebas limitando el *ancho de banda* que separa estos emplazamientos, para observar cómo se comporta el sistema implementado. En todos los casos, en el *Espacio de Trabajo Distribuido*, el *ancho de banda* de la WAN que separa los ERC y el ESCC será menor que en el caso de una LAN y, por tanto, que en la configuración de *Espacio de Trabajo Simple*. (Detalle en apdo. 8.2.3 Espacio de Trabajo Distribuido).

8.1.4 Aspectos a Considerar en el Entorno de Pruebas

Es importante destacar que el funcionamiento del sistema viene condicionado por varias variables importantes en una situación de producción.

Una de ellas es la resolución de los videos utilizados en la configuración de las *Ubicaciones*. A mayor resolución gráfica de los videos y las imágenes, se utilizará un mayor *ancho de banda* en transmitir el contenido, debido a que los archivos ocuparán mayor espacio físico en disco. La otra variable, muy a tener en cuenta en el *Marco*, es el *ancho de banda* disponible entre los diferentes equipos.

A mayor resolución del contenido multimedia, cuando se evalúa la aplicación de *Cartelería Digital*, se espera que se produzcan dos situaciones según la arquitectura implementada:

- Sin arquitectura Auto-Proxy: El equipo ERC solicitará un mayor *ancho de banda* cuando realice una conexión sobre el servidor para reproducir el video en modo *HTTP Streaming (Pseudo Streaming)*.
- Con arquitectura Auto-Proxy: Se establecerá una conexión que durará más en el tiempo para sincronizar el contenido desde el servidor en forma local.

Por tanto, el funcionamiento final del sistema podría verse afectado por la calidad de los videos y del *ancho de banda* disponible entre los emplazamientos físicos. Así, se ha de indicar que en este capítulo se presenta un entorno que permite diversos modos de funcionamiento de interés; aunque sin cubrir, como es evidente, la totalidad de situaciones posibles.

8.2 Esquema Lógico/Físico del Entorno de Pruebas

Para poder simular el funcionamiento de varios equipos (*ESCC*, *ERCs*), a partir de dos equipos portátiles, se han generado máquinas virtuales utilizando la aplicación *VMWare*. En concreto se ha realizado la instalación de los diferentes sistemas virtuales sobre una máquina anfitrión y, posteriormente, se han copiado al otro *equipo anfitrión*, las máquinas virtuales con el *software* desarrollado y las configuraciones específicas. De este modo, incluir más equipos en las pruebas es sencillo sin más que añadir nuevas máquinas virtuales en el entorno.

Parte del trabajo de implementación del presente proyecto se ha realizado sobre dichas *máquinas virtuales* estableciendo las aplicaciones y el código desarrollado (*swf*, *html*, *php*, *Scripts*,...) directamente en ellas, obteniendo una portabilidad muy flexible y pudiendo simular diferentes escenarios a partir de recursos *hardware* muy limitados.

Como contrapartida al uso de *máquinas virtuales* tenemos el inconveniente de que las prestaciones del equipamiento utilizado van a ser degradadas, al estar ejecutando el *sistema operativo* primario y diferentes *máquinas virtuales* de forma simultánea. Así, uno de los inconvenientes de las *máquinas virtuales* es que agregan gran carga al sistema en tiempo de ejecución. Esto tiene como efecto la ralentización del sistema; es decir, los programas desarrollados no alcanzarán la misma velocidad de ejecución que si se instalasen directamente en el *sistema operativo* primario o anfitrión (*host*), o directamente sobre una plataforma de

hardware. Sin embargo, dentro del presente *proyecto fin de carrera*, la flexibilidad que ofrece esta estrategia, compensa esta pérdida de eficiencia; además de permitirnos, a partir de *hardware* limitado, simular diferentes contextos de una mayor o menor complejidad.

Además del diseño, desarrollo y las pruebas de funcionamiento del *Marco* en su conjunto, se han preparado diferentes configuraciones lógicas a nivel de red para evaluar los distintos *Espacios de Trabajo* definidos en el Capítulo 4. En este sentido, se explorarán las diferentes situaciones con la modificación de la arquitectura de red desde escenarios muy simples, hasta escenarios típicos algo más complejos. Por tanto, la idea de verificar el funcionamiento del *Marco*, pasa por comprobar su funcionamiento en las diferentes posibles configuraciones de red. A continuación, se muestran los esquemas lógicos y físicos, para los que ha sido testeado el *Marco* en funcionamiento.

8.2.1 Espacio de Trabajo Simple Virtualizado

El esquema de trabajo que se presenta en la Ilustración 56 se ha establecido para realizar las primeras pruebas de funcionamiento sobre un mismo *equipo anfitrión*, evitando la necesidad de utilizar electrónica de red. Esta configuración también ha sido la utilizada para realizar el desarrollo del *Marco* y establecer las bases del funcionamiento.

El esquema se presenta en la siguiente ilustración y muestra, además de la configuración lógica, un acercamiento a la configuración física del entorno de pruebas:

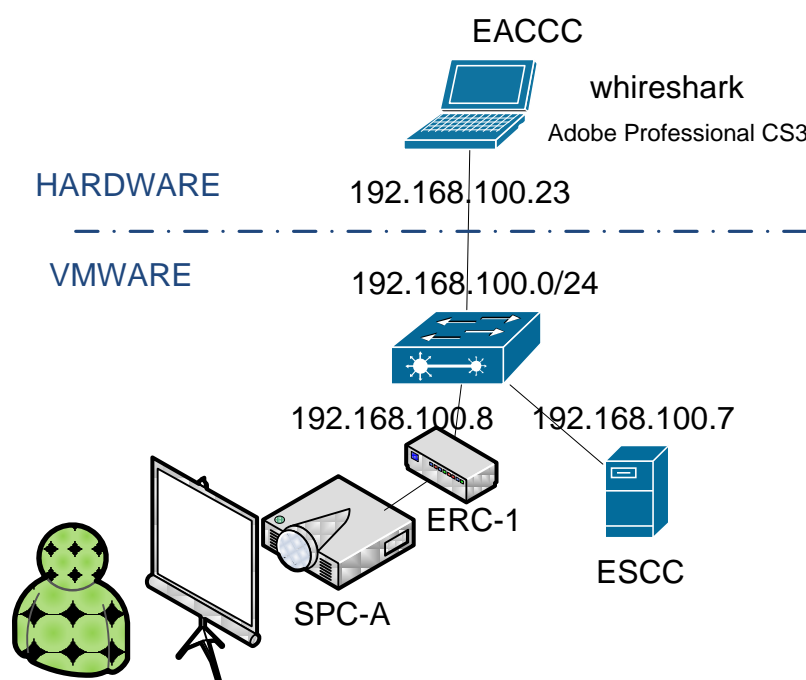


Ilustración 56.- Esquema lógico de configuración en Espacio de Trabajo Simple Virtualizado.

En el esquema anterior se indica cuál de los elementos de la arquitectura es *hardware* y cuál es un sistema *virtualizado*.

En esta ilustración se ha utilizado la conectividad desde el *equipo anfitrión* a través de uno de los interfaces virtuales del sistema (*vmnet1*), para acceder a la configuración de los equipos.

Por ello, aparece el *EACCC* representado como un equipo portátil como lógica de equipamiento *hardware*.

El resto de equipos se muestran como equipos virtualizados, ya que se han utilizado dos máquinas virtuales: una para el *ESCC* y otra para el *ERC*. Las modificaciones correctivas para las aplicaciones *swf* se han realizado desde el *equipo anfitrión* mediante la aplicación *Adobe Professional CS3* y, de hecho, ésta se ejecuta sobre el sistema operativo *Windows XP*.

8.2.2 Espacio de Trabajo Simple

En la Ilustración 57 se muestra un esquema lógico de la red implementada en el entorno de pruebas en la situación de *Espacio de Trabajo Simple*:

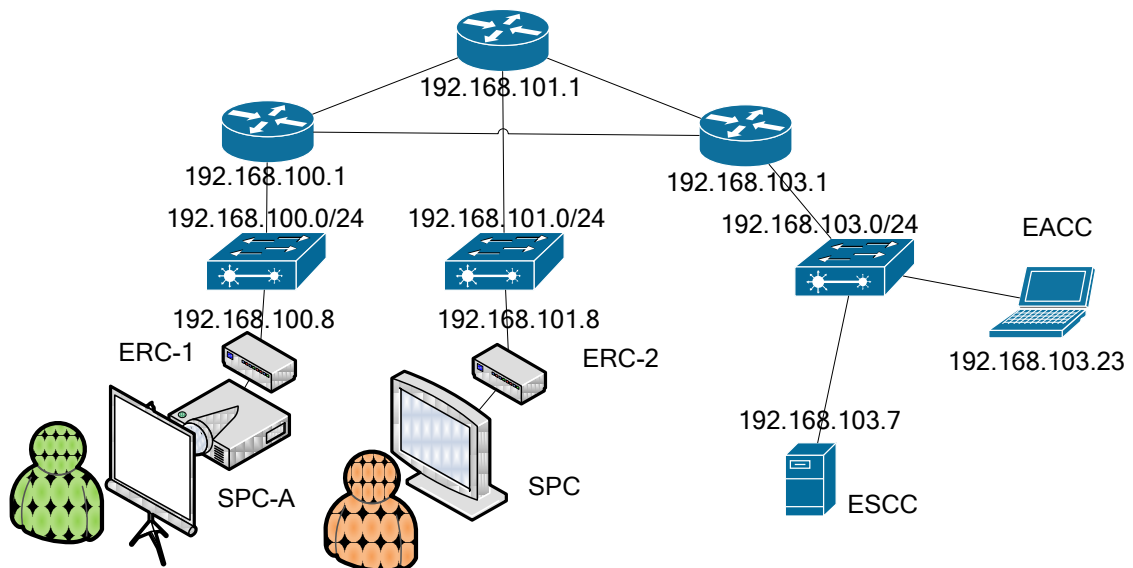


Ilustración 57.- Esquema Lógico del Entorno de Pruebas en la Configuración del Espacio de Trabajo Simple.

Sobre este esquema, simplemente resaltar, como se ha ido comentando a lo largo del proyecto, que el aspecto determinante que define un *Espacio de Trabajo Simple* o un *Espacio de Trabajo Distribuido* es la ausencia o presencia de una red *WAN* en el camino entre los *ERC* y el *ESCC*. Por tanto, la figura anterior representa un *Espacio de Trabajo Simple* ya que no se incluye una *WAN* entre los equipos.

En la Ilustración 58 se muestra el esquema lógico con detalle de la conexión física en el que se ha probado el sistema en una configuración de *Espacio de Trabajo Simple*:

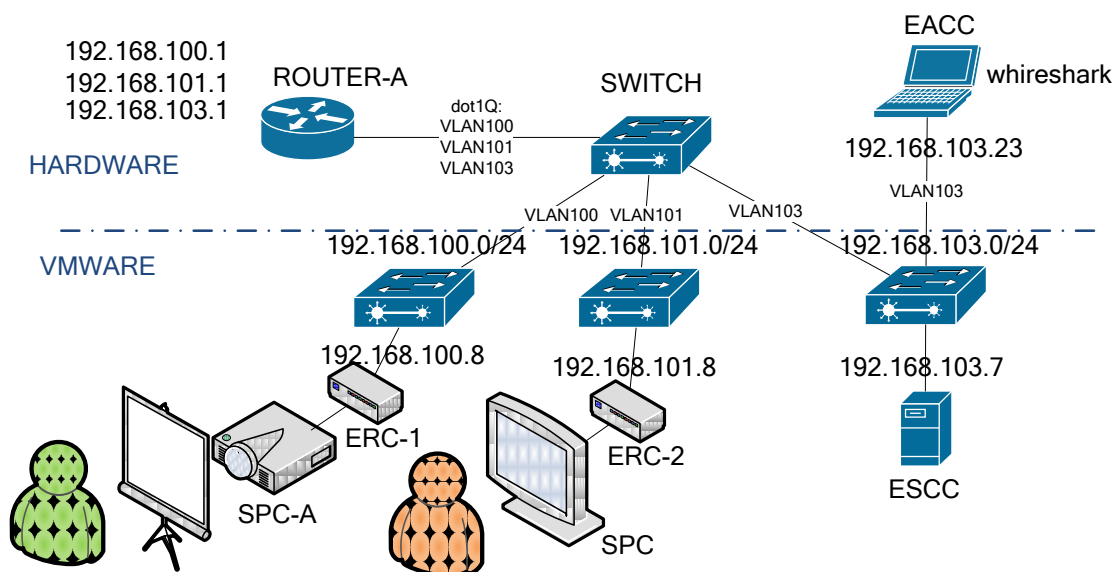


Ilustración 58.-. Esquema Lógico con detalle de implementación física del Entorno de Pruebas en la configuración del Espacio de Trabajo Simple.

Como se observa en la Ilustración 58, todos los equipos se mantienen dentro de varias redes *LAN* en las que no hay una red *WAN* entre los diferentes emplazamientos físicos. Se ha provisto de encaminamiento entre subredes en un equipo *router* para simular una situación que nos podemos encontrar habitualmente en ambientes de producción. Incluso es habitual que no exista el equipamiento *router*, pero de esta forma se configura de forma más genérica el *entorno de pruebas* y se evita tener que hacer apenas modificaciones al pasar a realizar pruebas más complejas.

A continuación, en la Ilustración 58, se muestra el *entorno de pruebas* representado en la Ilustración 57 en su representación más realista, mostrando las conexiones físicas establecidas en cada uno de los equipos. Se representan también las *VLANs* configuradas en los equipos:

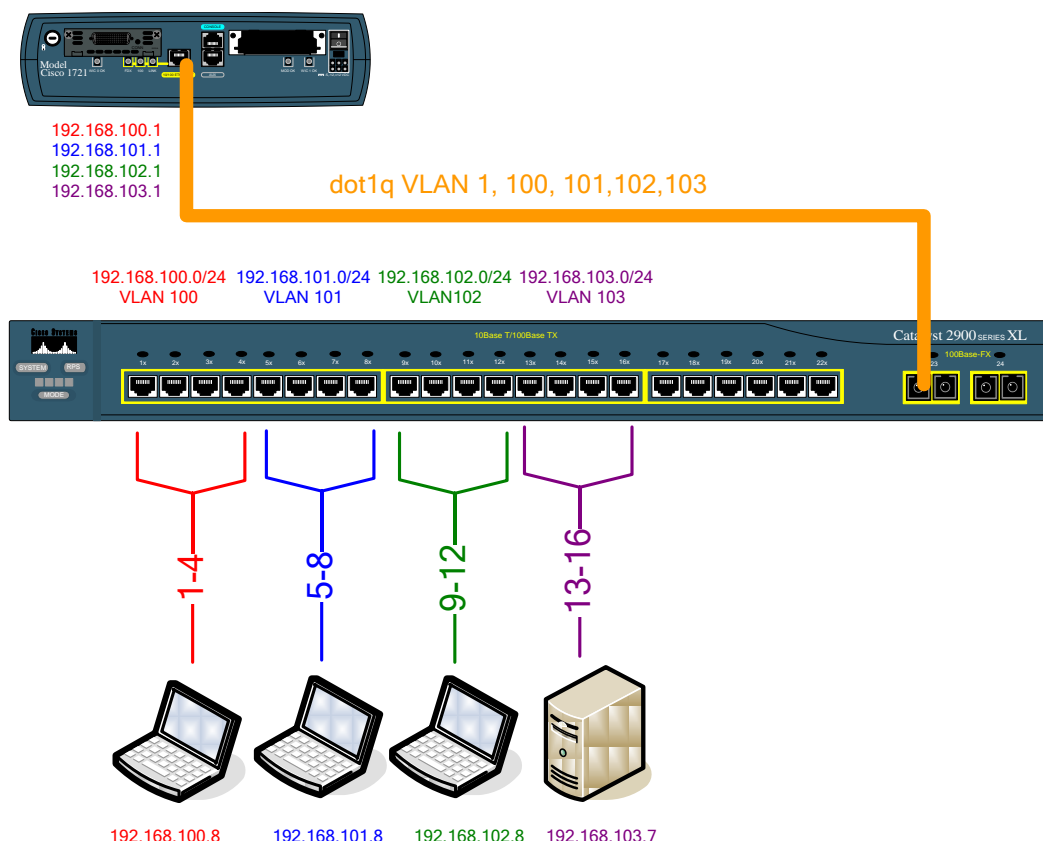


Ilustración 59.- Conexión física de equipos en Espacio de Trabajo Simple.

8.2.3 Espacio de Trabajo Distribuido

El *Espacio de Trabajo* más complejo, que se ha configurado para establecer las diferentes pruebas, es una arquitectura de red donde se introduce una *WAN* entre cada uno de los *ERCs* y el *ESCC*.

En esta situación se emula que los *ERC* se encuentren distribuidos en diferentes sedes de modo que se emula una *WAN* con un *ancho de banda* limitado por cada sede. Adicionalmente, el *ancho de banda* máximo de cada sede, debe estar limitado, además de por el *ancho de banda* configurado para cada sede, por el *ancho de banda* que la sede principal puede proporcionar y que es donde se encuentra el *ESCC*. Esta última, es una característica importante en el diseño de la red, ya que sería un error diseñar el *Marco* con un *ancho de banda* mínimo necesario para una sede, superior al que la sede principal puede proporcionar; si fuera así, podría verse afectado todo el sistema y no sólo la sede en cuestión mal dimensionada.

A continuación, se presenta (Ilustración 60) la configuración lógica de la red establecida en las pruebas para el *Espacio de Trabajo Distribuido*:

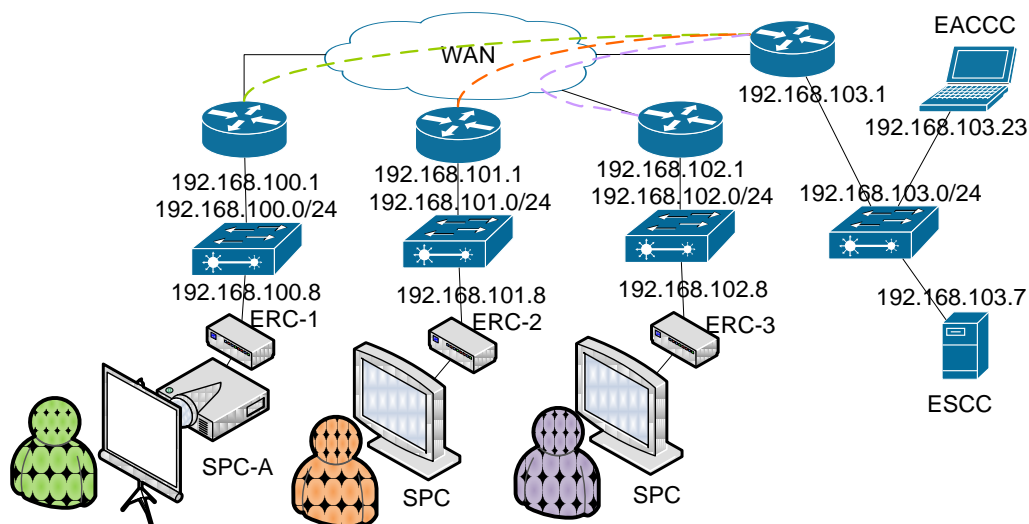


Ilustración 60.- Esquema lógico de Espacio de Trabajo Distribuido para 3 sedes más sede central.

Para emular la situación anterior de forma real, habría que disponer de diversos equipos de electrónica de red, además de conexiones con dichas sedes remotas. Para obtener un esquema como el que se presenta en la Ilustración 60, se ha establecido, como veremos, una estrategia donde se reduce de forma considerable el número de equipos para realizar las pruebas. Esto se obtiene haciendo uso, principalmente, de la *virtualización* de equipos y redes mediante configuración.

En la Ilustración 61 se muestra un esquema físico/lógico en el que se establece una representación incluyendo las conexiones físicas entre equipos. También se identifica como se implementa cada uno de los elementos: con equipamiento físico, mediante *virtualización* o VLANs. El esquema se muestra en la siguiente ilustración:

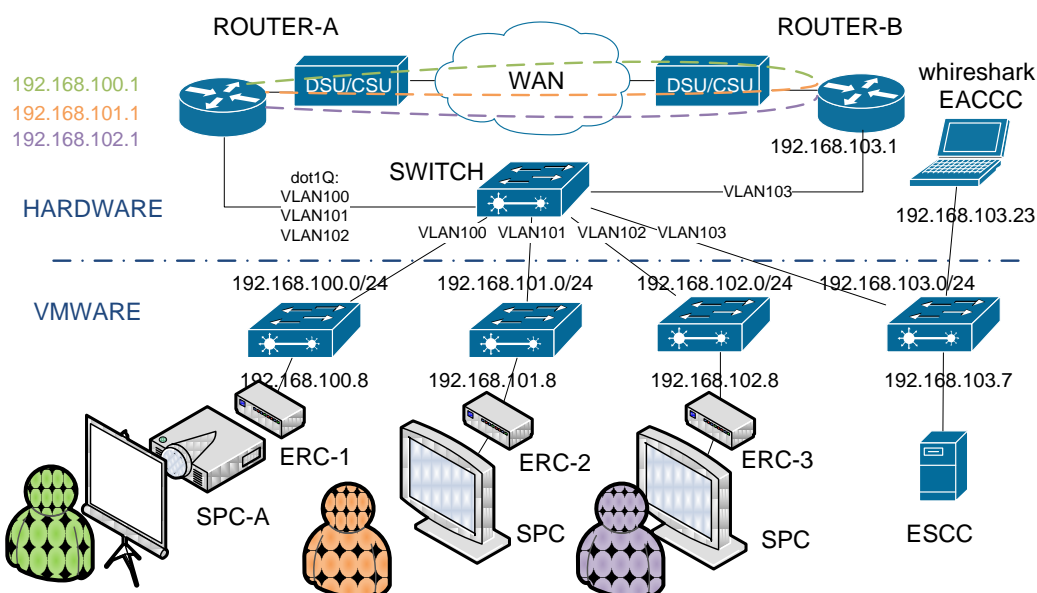


Ilustración 61.- Esquema Lógico con detalle de implementación Física del Entorno de Pruebas en la configuración del Espacio de Trabajo Distribuido.

Dentro del esquema (Ilustración 61), cada uno de los dispositivos lógicos se ha configurado mediante la combinación de diversos recursos: parte con *hardware*, parte con equipos *virtualizados*. Además, con la ayuda de configuraciones (*Sub-interfaces*, *Serial*, *rate-limit*, *ACL*) y protocolos (*dot1q*, *ppp*), se ha podido establecer la emulación de una *WAN* limitando el *ancho de banda* de subida y de bajada para cada hipotética sede y, a través de la configuración de *VLANs*, se ha podido establecer la arquitectura de red lógica que se muestra en la Ilustración 60. Las configuraciones de los equipos se pueden consultar en el APÉNDICE B Instalación y Configuración del Entorno de Pruebas. Con todo, con el esquema físico/lógico mostrado en la Ilustración 61, se obtiene el esquema lógico definido en la Ilustración 60.

El esquema físico que se presenta en la Ilustración 61, se ha obtenido con los equipos que se muestran en la Tabla 17 (apartado 8.1.1 Equipamiento Disponible para Entorno de Pruebas). La conexión física de estos dispositivos se muestra en la siguiente ilustración:

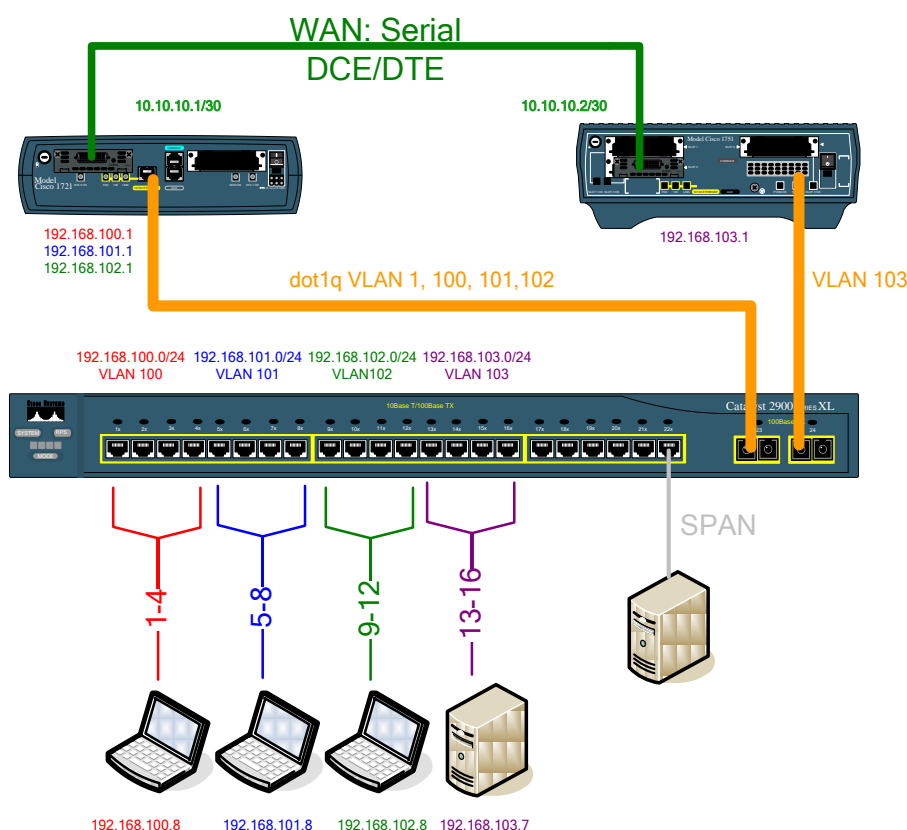


Ilustración 62.- Esquema Físico del Entorno de Pruebas en la configuración del Espacio de Trabajo Distribuido.

Se ha de notar la presencia un equipo en configuración *SPAN* (*Switched Port Analyzer*). Esta configuración, permite obtener sobre el puerto físico del *switch*, el tráfico producido de los diferentes equipos; sin embargo, las pruebas se han centrado en verificar el tráfico en la *VLAN103*, el cual corresponde con el tráfico del *ESCC* por lo que no se ha hecho, apenas, uso de esta posibilidad.

8.2.4 Conexión Física mediante AP

En algunas de las pruebas ha surgido la necesidad de conectar diferentes *ERCs* en distintas subredes. Se ha considerado conveniente separar el tráfico de cada subred en diferentes interfaces del *SWITCH*. De este modo se obtiene un entorno más realista y resulta más fácil el

análisis del tráfico capturado, ya que se elimina, en parte, la interacción de los protocolos de acceso al medio sobre una interfaz compartida. Para ello, la conexión de las *interfaces* de los equipos virtuales se debe realizar a diferentes puertos del *SWITCH* en la que haya configurada distintas *VLANs*.

Por otra parte, los equipos portátiles utilizados (*Equipo Anfitrión 1* y *Equipo Anfitrión 2*) no disponen de dos tarjetas de red tipo *Ethernet* para conectar los equipos virtuales a *VLANs* distintas. Sin embargo, estos equipos anfitriones disponen de tarjetas *wi-fi* y, además, se dispone de un dispositivo electrónico *wi-fi* (*FON2100A/B/C*), el cual se puede utilizar como un *AP* (*Access Point*). Por estas razones, se decidió establecer la configuración del dispositivo *FON2100A/B/C* como *AP*, para poder conectar los equipos anfitriones (y por ende los virtualizados) a través de las tarjetas *wi-fi* y el *SWITCH* a diferentes puertos con distintas *VLANs*.

La instalación de un nuevo *sistema operativo* sobre el dispositivo *FON2100A/B/C* queda fuera del alcance del proyecto; sin embargo, para establecer la configuración del *AP*, de una forma más versátil y libre, se cargó en el dispositivo un nuevo firmware³⁰ (el cual es una versión reducida del sistema operativo *Linux* para dispositivos embebidos):

- Firmware: *OpenWrt Kamikaze con X-Wrt Extensions 7.09*
- Kernel: *Linux 2.6.21.5*

La conexión física del *AP* utilizado con el resto de equipos, es la que se presenta en la siguiente ilustración:

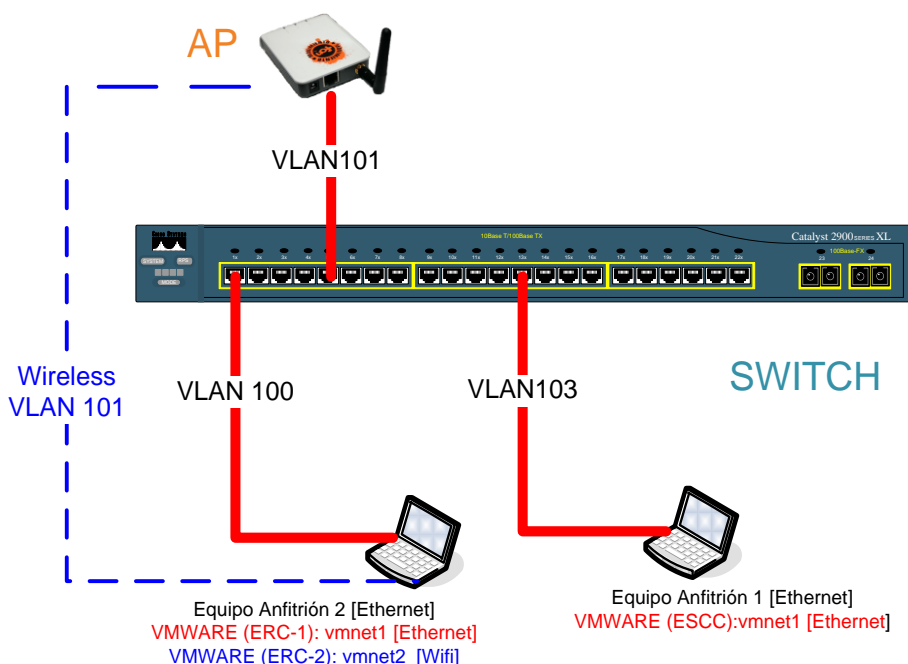


Ilustración 63.-. Esquema de conexión física de La Fonera, Equipos Anfitriones y SWITCH.

³⁰ Para instalar el firmware se siguieron las indicaciones que aparece en las URLs <http://www.fonera.info/index.php/topic,100.0.html> y <http://wiki.openwrt.org/toh/fon/fonera>. Para establecer el servidor *tftp*, se utilizó la aplicación *tftpd32* sobre el equipo anfitrión.

Esta conexión ha sido utilizada en varias de las pruebas. En el esquema sólo se muestra la conexión física entre los equipos anfitriones, el AP y el SWITCH; siendo la conexión lógica la establecida en cada uno de los tipos de pruebas.

Se puede observar que el *Equipo Anfitrión 2* mantiene la configuración de las tarjetas de red de cada uno de los equipos *virtualizados* de forma independiente. Por tanto, si en el SWITCH se configura la conexión del AP en una VLAN diferente (VLAN 101), a la conexión de la tarjeta Ethernet del *Equipo Anfitrión 2* (VLAN100), los ERCs (ERC-1 y ERC-2) virtualizados no tendrán conectividad si no intervienen elementos de nivel 3 en la arquitectura de red. Ocurre de igual modo con la configuración de la VLAN de la conexión física del ESCC (VLAN103). Así, utilizando esta estrategia e incorporando equipos *hardware* de nivel 3, se pueden obtener escenarios de pruebas de complejidad media y muy realista con solamente dos equipos portátiles y que nos permitirán evaluar el *Marco* con suficiente garantía.

Mediante la aplicación *iperf*, desde los equipos anfitriones, se evalúa el *ancho de banda* disponible cuando se utiliza una conexión cableada y cuando se utiliza la conexión mediante el AP conectado al SWITCH. El resultado del *ancho de banda* utilizando esta aplicación mediante las diferentes conexiones es el siguiente:

Tarjeta Red (ERC) --- SWITCH --- Tarjeta Red (ESCC)

```
[1868] local 192.168.103.23 port 5001
      connected with 192.168.103.32 port 1139
[ ID] Interval      Transfer    Bandwidth
[1868] 0.0-10.0 sec  107 MBytes  89.5 Mbits/sec
```

Tarjeta wi-fi (ERC) --- AP --- SWITCH --- Tarjeta Red (ESCC)

```
[1856] local 192.168.103.23 port 5001
      connected with 192.168.103.32 port 1217
[ ID] Interval      Transfer    Bandwidth
[1856] 0.0-10.0 sec  23.1 MBytes  19.4 Mbits/sec
```

Como se observa la diferencia es considerable y se deberá tener presente en alguna de las pruebas para interpretar de forma correcta el tráfico capturado.

8.2.5 Espacio de Trabajo Distribuido en Emisión Streaming Media Local

Las pruebas realizadas sobre el *Marco*, introduciendo como mejora la posibilidad de recuperar contenido multimedia mediante un flujo *Multicast*, han estado orientadas a comprobar el funcionamiento de la nueva funcionalidad en su conjunto. Adicionalmente, se han capturado algunos flujos de red para verificar las variaciones del uso del *ancho de banda*.

El esquema configurado ha sido el mismo que el utilizado para el *Espacio de Trabajo Distribuido*, pero se ha introducido un nuevo equipo (ESVLC) como se ha mostrado en la Ilustración 51.

A través de la tarjeta de red de uno de los equipos anfitriones, se realiza, por tanto, la emisión del flujo RTP/MPEG-TS en modo *Multicast*. El esquema se muestra en la siguiente ilustración:

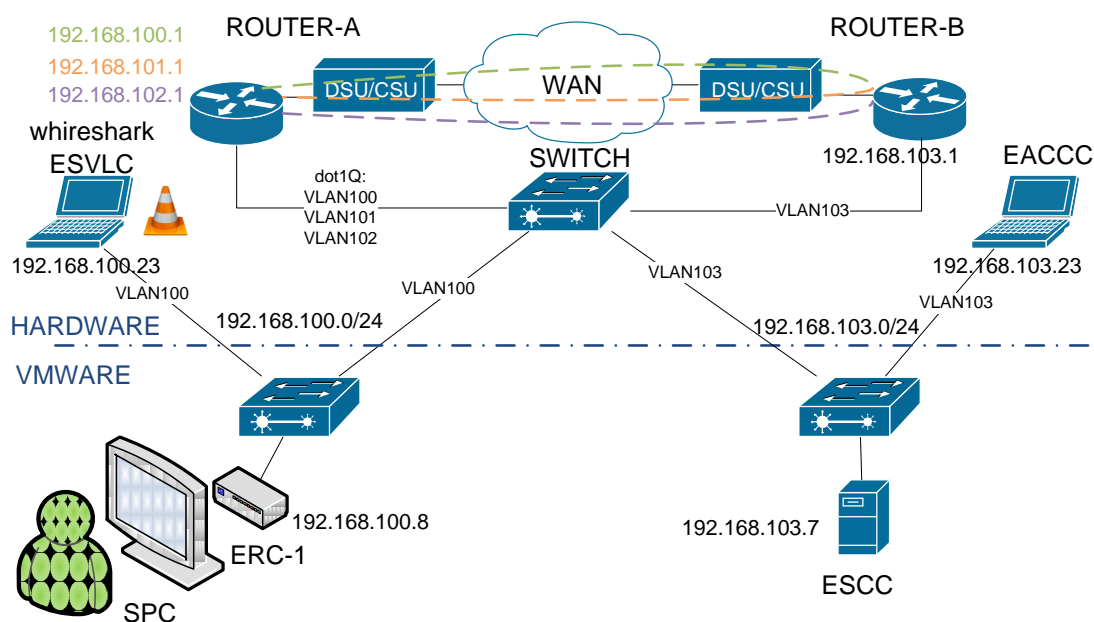


Ilustración 64.-. Esquema Lógico con detalle de implementación Física del Entorno de Pruebas en la configuración del Espacio de Trabajo Distribuido para pruebas de Mejoras Introducidas.

Para las pruebas de funcionamiento se ha utilizado únicamente una sede remota según el esquema de la Ilustración 64 y se ha incluido un nuevo equipo (*ESCVLC*). Este nuevo equipo se obtiene, directamente, configurando la aplicación *VLC* en el *equipo anfitrión* y conectando el interfaz de red *Ethernet* al *SWITCH*. Así, la conexión física y la configuración de los equipos de electrónica de red es exactamente la misma que la utilizada en el apartado 8.2.3. De este modo, combinándola con el uso del *AP* para el *ERC*, cuya conexión se representa en el apdo. 8.2.4, se obtiene el esquema de la Ilustración 64. Tanto la tarjeta *Ethernet* del *equipo anfitrión*, como el *AP* que enlaza con *vmnet1* de forma inalámbrica, están conectados a la misma *VLAN* del *SWITCH* (*VLAN100*); no obstante, los tráfico generados y recibidos están separados en cada puerto del *SWITCH*.

A continuación, se muestra una imagen real de los equipos configurados para establecer los *espacios de trabajos* definidos utilizando, también, el equipo *FON2100A/B/C* como *AP*:



Ilustración 65.-. Conexión Física del Entorno de pruebas para todos los espacios de trabajo definidos.

8.3 Tipos de Pruebas Realizadas

Fundamentalmente, las pruebas ideadas consisten en establecer las conexiones en diferentes situaciones y ver cómo se comporta el sistema a nivel de congestión de la red (forma *cuantitativa*) y, de forma individual y subjetiva, verificar en la pantalla de presentación (forma *cualitativa*) de cada *ERC* que está emitiendo las configuraciones realizadas.

En las pruebas, se ha forzado el *Marco* a que trabaje en diferentes situaciones que se han considerado de interés. En concreto el *Marco* que se ha desarrollado implementa la característica definida como *Auto-Proxy*, por lo que se forzará para que trabaje como si ésta no se hubiera implementado. Es decir, compararemos el *Marco* con otro de iguales características, pero implementando otra arquitectura funcional más simple. De este modo se podrá evaluar, con el resultado del funcionamiento en las distintas *arquitecturas funcionales*, cómo afecta la implementación de esta característica (*Auto-Proxy*) al *Marco* desarrollado.

La herramienta *crontab*, la cual se ha configurado para que lance la sincronización mediante un *script* a intervalos regulares, se desactiva en toda la fase de pruebas. Realizando este procedimiento se consigue evaluar el *Marco* desarrollado sin la intervención en el sistema de una *arquitectura Auto-Proxy* debido a que hasta que no se reciba la orden de sincronización en el *ERC*, éstos no sincronizarán el contenido multimedia de forma local con el *ESCC*. De este modo, para lograr la sincronización de archivos en las pruebas realizadas, se hará uso de la *interfaz gráfica de usuario* programada para establecer la sincronización de contenido en los *ERCs*.

8.3.1 Espacio de Trabajo en modo NAP/NS (No Auto-Proxy y No Sincronización)

Con este modo forzado para la *arquitectura funcional* se pretende emular el *Marco* como si no se hubiera implementado la característica *Auto-proxy*.

Para establecer este funcionamiento en la arquitectura desarrollada del sistema se sigue la siguiente pauta, una vez se ha logrado la conectividad entre equipos:

- Se configura el *ESCC* con la reproducción de archivos en las *listas de reproducción* escogida para la *dirección IPv4* del *ERC*.
- Se enciende/n el/los *ERC/s* de modo que se espera a que finalice el proceso de sincronización de contenido de forma local. La tarea *crontab* para la sincronización de contenido está desactivada.
- Se borran manualmente³¹ (sin intervención de ninguna herramienta web de administración) los archivos que se hayan almacenado en los *ERCs* de forma local.
- Se establecen órdenes de actualización para las *ubicaciones* a través del *servidor socket* desde la administración del *ESCC*.

De este modo, los *ERCs* no tienen contenido multimedia almacenado de forma local y se inicia la petición de configuración al *ESCC*. Debido a que los *ERCs* no encuentran contenido multimedia almacenado de forma local, se establece un flujo *HTTP Streaming* desde el *ESCC* hacia cada *ERC*, obteniendo el *ERC* el funcionamiento correcto, pero sin aplicar la arquitectura *Auto-Proxy*.

8.3.2 Espacio de Trabajo en modo NAP/CS (No Auto-Proxy y Con Sincronización): Peor Situación

En este modo de funcionamiento forzado de *arquitectura funcional*, se pretende emular el sistema para que se muestre el *Marco* desarrollado en un estado de peor situación posible.

La situación más desfavorable del *Marco*, de todas las posibles con la arquitectura *Auto-Proxy* implementada, es que todos los equipos *ERCs* obtengan, de forma simultánea por primera vez, conectividad con el *ESCC*. Así, establecen el inicio de la sincronización de archivos contra el servidor de forma simultánea y, adicionalmente, estén realizando *HTTP Streaming* sobre los archivos allí almacenados.

Para lograr este comportamiento forzado sobre el *Marco* desarrollado, se establecen los siguientes pasos:

- Se establece el procedimiento para configurar el sistema en situación de configuración del *Marco* sin arquitectura *Auto-Proxy* en todos los *ERCs*; tal y como se describe en el sub-apartado anterior.
- Una vez borrado los archivos multimedia de todos los *ERCs* y con todos los equipos conectados al *socket*, indicamos desde el *ESCC* la *Orden de Sincronización* global (mediante la *interfaz gráfica de usuario*) para que todos los *ERCs* comiencen la sincronización al mismo tiempo y de forma simultánea (la tarea *crontab* se mantiene desactivada). Cada uno de ellos utiliza el *ancho de banda* configurado para tal fin (parámetros de configuración de cada *ERC*) y disponible según la *WAN*, la cual será función del emplazamiento físico en que se encuentre el equipo.

Con lo anterior se deben observar dos tipos de orígenes de tráfico simultáneo desde el *ESCC* hacia cada *ERC*. Uno, el relacionado con la *sincronización* de forma local del contenido multimedia con el *ESCC* y el otro, el relativo al *HTTP Streaming* establecido desde el *ESCC* a cada *Zona* configurada de cada *ERC*.

³¹ Se establece una conexión *ssh* hacia el *ERC* y se borran los archivos multimedia almacenados en los directorios locales.

8.3.3 *Espacio de Trabajo en modo AP/CS (Auto-Proxy y Con Sincronización)*

Esta situación puede obtenerse a partir de la situación estable del sistema, en la que el comportamiento es *Auto-proxy*, y, sin embargo, se ha subido nuevo contenido multimedia al servidor. De este modo y por funcionalidad de la aplicación, cada vez que se adjunta un nuevo archivo al servidor, se establece la orden de sincronización a todos los *ERCs*.

En esta situación no se debe observar tráfico *HTTP Streaming* desde el *ESCC* hacia los *ERC*.

8.3.4 *Espacio de Trabajo en modo AP/NS (Auto-Proxy y No Sincronización): Modo Estable*

Finalmente, la última forma de trabajar de la *arquitectura funcional* evaluada será cuando el sistema se mantiene funcionando de forma estable; es decir, cuando todos los *ERCs* han actualizado el contenido multimedia de forma local y, por tanto, recuperan los archivos a difundir desde el propio *ERC*.

En esta situación no habrá por tanto ni tráfico *HTTP Streaming* desde el *ESCC*, ni tráfico debido a la sincronización con los equipos.

8.4 *Resumen*

En este capítulo se ha definido y mostrado las conexiones sobre los equipos que nos permite trabajar en un *entorno de pruebas* muy realista. Este *entorno de pruebas*, el cual se ha construido con equipamiento *hardware* muy limitado, nos permite simular diferentes contextos similares a los que nos podemos encontrar en las típicas situaciones de producción en las organizaciones que dispongan de una red de comunicación.

Para la elaboración y generación del *entorno de pruebas* de forma rigurosa y realista se ha hecho uso de la combinación de varias herramientas *software* (*vmware*, *sistemas operativos* y configuración del *ios* de los *routers*) y diverso equipamiento de red *hardware* (*routers*, *switches*, *ordenadores personales*, *puntos de acceso inalámbrico*). Esta combinación ha permitido reducir el número de elementos necesarios para generar un *entorno de pruebas* apropiado y ha permitido reducir los costes del proyecto en lo relativo a la evaluación del *Marco*.

Para la evaluación del *Marco* se ha utilizado *software* adicional (*vlc*, *Wireshark*, *iperf*, *Origin*, *Media Info*) que nos permite medir las características del *Marco* de modo cuantitativo. También se ha mostrado algunas consideraciones a tener en cuenta en la evaluación y el método utilizado para simular *arquitecturas funcionales* más simples (sin la característica *Auto-Proxy*) a partir de la manipulación manual del *Marco* desarrollado. La ejecución de la aplicación de *Cartelería Digital* sin la característica *Auto-Proxy* nos permitirá comparar esta arquitectura más simple con respecto al *Marco* desarrollado que sí la implementa esta característica en la *arquitectura funcional*.

Capítulo 9 Resultados de las Pruebas Realizadas

En este capítulo se describen los resultados de las pruebas más representativas que se han realizado en el Entorno de Pruebas presentado en el Capítulo 8. Se ha de considerar que en todas las configuraciones se tienen diferentes *ERCs* (*Equipos Reproductores Cliente*) *virtualizados*, con conectividad al *ESCC* (*Equipo Servidor de Contenido y Configuración*) también *virtualizado*. Este entorno se obtiene con una adecuada configuración de los diferentes equipos *hardware* y *software* utilizados para la elaboración del *entorno de pruebas* (ver Capítulo 8 Entorno de Pruebas y ANEXO B: Instalación y Configuración del Entorno de Pruebas).

Se hicieron pruebas funcionales del *Marco* para comprobar su correcto funcionamiento. Para que el lector pueda verificar el comportamiento adecuado del mismo, se han realizado algunas grabaciones de vídeos que se han publicado en el canal <http://www.youtube.com/user/dtimonPFC>. En este capítulo nos centramos en el estudio de los efectos que las distintas configuraciones del *Marco* tienen en el tráfico generado por el funcionamiento del mismo.

9.1 Pruebas de Funcionamiento del Marco para Aplicación de Cartelería Digital y Técnica HTTP Streaming

Para la realización de las pruebas se han utilizado diferentes *listas de reproducción*, con diferentes *archivos multimedia* de diferentes características, las cuales se han asignado a cada una de las *Zonas* configuradas en cada *Ubicación*. Debido a que las *listas de reproducción* se pueden asociar a cada *Zona* del *Escenario* de cada *Ubicación*, se presenta un listado inicial de las *listas de reproducción* utilizadas para presentar la secuencia de archivos de reproducción en las pruebas:

Lista Reproducción		Escenario	N Archivos	BW Necesario HTTP Streaming
Lista_1	trailers	0	2	2.286 Kbps
Lista_2	cultClientPruebLQ	0	2	296 Kbps
	cultClientPruebMQ	1		517 Kbps
	cultClientPruebHQ	2		679 Kbps
Lista_3	genUc3mPruebLQ	0	2	314 Kbps
	genUc3mPruebMQ	1		575 Kbps
	genUc3mPruebHQ	2		1.050 Kbps

Tabla 19.-. Necesidades mínimas de *ancho de banda* en Listas de Reproducción Utilizadas en las pruebas según aplicación MediaInfo.

El *ancho de banda* (BW^{32}) *Necesario* en cada *lista de reproducción* se calcula como el mínimo *ancho de banda* necesario para reproducir el archivo en *HTTP Streaming*, sin que se produzcan cortes en la reproducción. Este valor, presentado en la tabla anterior, obedece a los archivos a reproducir y dependen de parámetros tales como la resolución del vídeo o la calidad en la codificación del vídeo y audio, además del *contenedor multimedia* utilizado para la generación del archivo (siempre *.flv*).

En la siguiente tabla se presenta un listado con algunas de las principales características de los archivos utilizados:

³² Del inglés *BandWidht* (*Ancho de Banda*).

Archivo /ZONA ³³ /ESC	Tipo	Resolución/Ratio/códec Frame x seg/flujo (BW)	Muestreo/bits/canale s códec (BW)	BW Total (HTTP Streaming)
Lista_1 (Nombre: trailers/Ubicación: cines_hd)				
1/A/0	VÍDEO	1280x720/16:9/H.263 25fps/constante (2.058Kbps)	44,1KHz/16bit/2 MPEG Audio v1/constante/LAME3.9 7 (128Kbps)	2,286Kbps (286KBytes/s) 02:10 min: seg 35,6 MBytes
2/A/0	VÍDEO	320x262 /~5:4/H.263 -/variable (325 Kbps)	22,05KHz/16bit/1 MPEG Audio v2/variable	325Kbps (40KBytes/s) 45 seg 1,76 MBytes
Lista_2_low (Nombre: cultCientPruebLQ/Ubicación: cultCient/ESC:0) Lista_2_medium (Nombre: cultCientPruebMQ/ Ubicación: cultCient/ESC:1) Lista_2_high (Nombre: cultCientPruebHQ/ Ubicación: cultCient/ESC:2)				
1/A/0	VÍDEO	300x240/5:4/H.263 25fps/constante (248Kbps)	22,05KHz/16bit/2 MPEG Audio v2 (L3)/variable (40,6Kbps)	296Kbps (37KBytes/s) 4min 9,77MB
1/A/1	VÍDEO	600x480/5:4/AVC(Main@L3.0) 25fps (YUV)/constante (464Kbps)	44,1KHz/16bit/2 AAC(v4; LC) (48,5Kbps)	517Kbps (65KBytes/s) 4min 17,1MB
1/A/2	VÍDEO	600x480/5:4/AVC(Main@L3.1) 29,970fps (YUV)/constante 535Kbps	44,1KHz/16bit/2 AAC(v4; LC) (96,6Kbps)	638Kbps (78KBytes/s) 4min 21,1MB
2/A/0	VÍDEO	300x240/5:4/H.263 25fps/constante (248Kbps)	22,05KHz/16bit/2 MPEG Audio v2 (L3)/variable (40,3Kbps)	296Kbps (37KBytes/s) 3min 7,07MB
2/A/1	VÍDEO	450x360/5:4/AVC(Main@L2.1) 25fps (YUV)/constante (355Kbps)	44,1KHz/16bit/2 AAC(v4; LC) (48,5Kbps)	408Kbps (51KBytes/s) 3min 9,77MB
2/A/2	VÍDEO	600x480/5:4/AVC(Main@L3.1) 25fps (YUV)/constante 576Kbps	44,1KHz/16bit/2 AAC(v4; LC) (96,6Kbps)	679Kbps (85KBytes/s) 3min 21,1MB
Lista_3_low (Nombre: genUc3mPruebLQ /Ubicación: genUc3m/ESC: 0) Lista_3_medium (Nombre: genUc3mPruebMQ/ Ubicación: genUc3m/ESC: 1) Lista_3_high (Nombre: genUc3mPruebHQ/ Ubicación: genUc3m/ESC: 2)				
1/A/0	VÍDEO	400x226/16:9/H.263 29,970fps/constante (256Kbps)	22,05KHz/16bit/2 MPEG Audio v2 (L3)/variable (8000bps)	295Kbps (37KBytes/s) 6min 14,6MB
1/A/1	VÍDEO	640x360/16:9/AVC(Main@L3.0) 29,970fps (YUV)/constante (417Kbps)	44,1KHz/16bit/2 AAC(v4; LC) (47,7Kbps)	471Kbps (59KBytes/s) 6min 23,3MB
1/A/2	VÍDEO	854x480/16:9/AVC(Main@L3.1) 30fps (YUV)/constante (910Kbps)	44,1KHz/16bit/2 AAC(v4; LC) (95,4Kbps)	1 013Kbps (127KBytes/s) 6min 50,2MB
2/A/0	VÍDEO	400x226/16:9/H.263 29,970fps/constante (246Kbps)	22,05KHz/16bit/2 MPEG Audio v2 (L3)/variable (59,3Kbps)	314Kbps (39KBytes/s) 3min 7,22MB
2/A/1	VÍDEO	640x360/16:9/AVC(Main@L3.0) 29,970fps (YUV)/constante 460Kbps	44,1KHz/16bit/2 AAC(v4; LC) (108Kbps)	575Kbps (72KBytes/s) 3min 13,3MB

³³ Cada Zona consume un *ancho de banda* específico y condicionado por los archivos programados en la lista asociada a la Zona. Sin embargo, cuando se reproducen dos zonas o más, sobre una *Ubicación*, se ha de considerar que el *anchos de banda* utilizado es debido a la suma de los flujos *HTTP Streaming* de cada Zona.

Archivo /ZONA ³³ /ESC	Tipo	Resolución/Ratio/códec Frame x seg/flujo (BW)	Muestreo/bits/canale s códec (BW)	BW Total (HTTP Streaming)
2/A/2	VÍDEO	854x480/16:9/AVC(Main@L3.1) 29,970fps (YUV)/constante 936Kbps	44,1Khz/16bit/2 AAC(v4; LC) (108Kbps)	1 050Kbps (131KBytes/s) 3min 24,3MB

Tabla 20.-. Características de los archivos utilizados en las listas de reproducción configuradas en las pruebas.
Datos obtenidos por la aplicación MediaInfo.

En la Tabla 20 se muestra los parámetros principales de los archivos utilizados en las *listas de reproducción* que se ha programado para realizar las pruebas. Se han creado diversas *listas* con los mismos *archivos* a reproducir en diferentes calidades según la codificación del vídeo. La notación utilizada, por tanto, para las *listas de reproducción*, se adopta en función de los archivos multimedia que contiene:

- *LQ, Low Quality.* Calidad Baja
- *MQ, Medium Quality.* Calidad Media
- *HQ, High Quality.* Calidad Alta

La notación utilizada para nombrar las distintas pruebas ha sido:

<0X00Y>

Donde X representa un bloque de pruebas comunes en el que los parámetros de red y configuración del *Marco* no varían; e Y hace referencia al modo de trabajo forzado del *Marco*, en el cual se ha considerado los siguientes:

00Y	Notación ³⁴	Auto-proxy	rsync	Descripción del Modo de Funcionamiento del Marco
0X001	NAP/NS	No	No	Sin Auto-Proxy
0X002	NAP/CS	No	Sí	Sin Auto-Proxy y Sincronizando: <u>Peor situación</u>
0X003	AP/CS	Sí	Sí	Con Auto-Proxy y Sincronizando
0X004	AP/NS	Sí	No	Con Auto-Proxy y Estable: <u>Mejor situación</u>

Tabla 21.-. Notación de Pruebas para la evaluación de la aplicación de Cartelería Digital sobre el Marco y para la evaluación de característica Auto-Proxy sobre la arquitectura funcional.

Se han ido ajustando algunos de los parámetros de de los distintos *espacios de trabajo* evaluados, para establecer un pequeño estudio del impacto sobre el *Marco* de cambios en la configuración de la red.

En la Tabla 22, Tabla 23 y Tabla 24 se muestran las características principales del *entorno de pruebas* en cada una de los bloques de pruebas realizados. Algunas de las características presentadas en estas tablas son las siguientes:

- *Tipo de Espacio de Trabajo:* si es un *Espacio de Trabajo Simple o Distribuido*.
- *ERCs:* Corresponde con el número de *Equipos Reproductores Clientes* habilitados en la prueba.

³⁴ Ver apartado 8.3Tipos de Pruebas Realizadas.

- *BW WAN ESCC*: *Ancho de Banda WAN* configurado en la sede principal donde se encuentra el *ESCC*. La configuración se realiza mediante el enlace *Serial* de los *routers* del *entorno de pruebas* (ver *Capítulo 8* y ANEXO B).
- *BW rsync*: *Ancho de Banda* configurado en cada *ERC* para la aplicación *rsync*. Esta aplicación es la encargada de sincronizar con el *ESCC*, el contenido multimedia de forma local en cada *ERC*.
- *BW WAN ERC*: Es el *ancho de banda* configurado en cada sede remota donde se encuentran los *ERCs*. En cada *sub-interfaz* del *Router A* es posible limitar el *ancho de banda* (ver *Capítulo 8* y ANEXO B).

En la Tabla 22 se muestran los valores de las configuraciones principales del *entorno de pruebas* en el estudio del funcionamiento del *Marco* para la aplicación de *Cartelería Digital* de acuerdo a la notación presentada en la Tabla 21.

Nombre	Tipo ³⁵	Tipo Espacio de Trabajo	ERCs	BW WAN ESCC	BW Rsync	BW WAN ERC
Espacio de Trabajo Simple Virtualizado: 0100X (pág. 223)						
01001	NAP/NS	Simple	1	N/A (virtualizado)	4.194.304 bps 512 KB/s	N/A (virtualizado)
01002	NAP/CS					
01003	AP/CS					
01004	AP/NS					
Espacio de Trabajo Simple con Conexión Física. 0200X (pág. 229)						
02001	NAP/NS	Simple	1	LAN (wi-fi) [19,4 Mbps]	4.194.304 bps 512 KB/s	LAN (wi-fi) [19,4 Mbps]
02002	NAP/CS					
02003	AP/CS					
02004	AP/NS					
Espacio de Trabajo Distribuido con 1 solo Equipo ERC. 0300X (pág. 233)						
03001	NAP/NS	Distribuido	1	4.000.000 bps [3,81 Mbps] 488 KB/s (Serial)	1.048.576 Kbps 128 KB/s	4.000.000 bps [3,81 Mbps] 488 KB/s
03002	NAP/CS					
03003	AP/CS					
03004	AP/NS					
Espacio de Trabajo Distribuido con 2 Equipos ERC. Configuración límite del BW entre sedes. 0400X (pág. 235)						
04001	NAP/NS	Distribuido	2	8.000.000 bps [7,63 Mbps] 977 KB/s (Serial)	2.097.152 bps 256 KB/s (ERC-1/ERC-2)	4.000.000 bps [3,81 Mbps] 488 KB/s (ERC-1/ERC-2)
04002	NAP/CS					
04003	AP/CS					
04004	AP/NS					
Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN Insuficiente en una sede: 0500X. (pág. 238)						
05001	NAP/NS	Distribuido	2	8.000.000 bps [7,63 Mbps] 977 KB/s (Serial)	2.097.152 bps 256 KB/s (ERC-1/ERC-2)	1.048.576 kbps 128 KB/s (ERC-1) 2.097.152 bps 256 KB/s (ERC-2)
05002	NAP/CS					
05003	AP/CS					
05004	AP/NS					
Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN necesario en la sede del ESCC. 0600X. (pág. 241)						
06001	NAP/NS	Distribuido	2	1.000.000 bps [976 Kbps] 122 KB/s (serial)	1.048.576 kbps 128 KB/s (ERC-1/ERC-2)	4.194.304 bps [4 Mbps] 512 KB/s (ERC-1/ERC-2)
06002	NAP/CS					
06003	AP/CS					
06004	AP/NS					

Tabla 22.-. Resumen de las principales pruebas realizadas para la evaluación de la aplicación de Cartelería Digital desarrollada sobre el Marco.

También, se presenta el resumen de la prueba realizada para la evaluación de mejora presentada para establecer la recepción y reproducción de un flujo *Streaming Media Multicast*.

Nombre	Tipo Espacio de Trabajo	ERCs	BW WAN ESCC	BW Rsync	BW WAN ERC
Prueba de Funcionamiento del Marco para Aplicación de Difusión Selectiva con Flujo Streaming Media Multicast (pág. 243)					
Mejoras: Multicast	Simple	1	8.000.000 bps [7,63 Mbps] 977 KB/s (Serial)	N/A	LAN (wi-fi) [19,4 Mbps]

Tabla 23.-. Resumen de las pruebas realizadas para la evaluación de la mejora Streaming Media Multicast sobre el Marco.

Por último, Se presentan también algunas pruebas que aplican a todos los bloques de pruebas, por lo que se han denominado *Pruebas Transversales*. A continuación, se presenta la configuración de red establecida para las de las pruebas transversales realizadas:

³⁵ Ver apartado 8.3 Tipos de Pruebas Realizadas.

Nombre	Tipo Espacio de Trabajo	ERCs	BW WAN ESCC	BW Rsync	BW WAN ERC
<ul style="list-style-type: none"> • Prueba Transversal: Secuencia de Inicio del ERC. T001 (pág. 245) • Prueba Transversal de Memoria Caché del Navegador. T002. (pág. 252) • Prueba Transversal Subjetiva de Calidad de Imagen en función de la resolución y del ancho de banda necesario en una sede. T003. (pág. 256) 					
T001 T002 T003	Simple	1	LAN (Ethernet) [89,5 Mbps]	512 KB/s	LAN (Ethernet) [89,5 Mbps]

Tabla 24.-. Resumen de las pruebas transversales realizadas.

Existen otras estrategias de adquisición de datos como la presentada en la Ilustración 62, mediante la configuración del *SWITCH* (puerto en modo *SPAN*) y la conexión de un equipo para que obtenga todos los flujos correspondientes a cada *VLAN*; no obstante, debido a que, para obtener el tráfico de esta forma, es necesario utilizar una tarjeta de red de forma exclusiva en uno de los equipos se opta por adquirir los datos en la tarjeta utilizada en el propio *ESCC* (*equipo anfitrión*).

Se ha de indicar que una situación normal de funcionamiento del *Marco* para la aplicación de *Cartelería Digital*; es decir, con *Auto-Proxy* y *No Sincronización (AP/NS)* de archivos, no se detecta tráfico en la red relativo a la transmisión de los archivos en todas las pruebas realizadas, de modo que el tráfico intercambiado es el relativo a las peticiones/envío de configuración de los equipos. Por esta razón no se muestra el resultado gráficamente en el modo *AP/NS*.

A continuación, en los sucesivos sub apartados, se muestra el resultado de las pruebas realizadas. Fundamentalmente se presenta el resultado mostrando el *ancho de banda* utilizado en función del tiempo. Estas gráficas se obtienen mediante la adquisición del tráfico en el *ESCC* con la aplicación *Wireshark* y se presentan utilizando la aplicación *Origin Pro 8*.

9.1.1 Espacio de Trabajo Simple Virtualizado: 0100X

De acuerdo con las pruebas presentadas en la Tabla 22, la codificación del grupo de pruebas *0100X* corresponde con las realizadas en un *Espacio de Trabajo Simple* en el que se han *virtualizado* tanto el *ERC* como el *ESCC* sobre el mismo *equipo anfitrión*. En el bloque de pruebas *0100X*, la interfaz de red de cada equipo emulado es compartida y también *virtualizada*, de modo que se hace uso de la interfaz *vmnet1* definida. Esta configuración también ha sido la utilizada para desarrollar el *Marco* diseñado, de modo que en la fase de desarrollo sólo ha sido necesaria la utilización de un equipo físico anfitrión.

Para la realización de este bloque de pruebas se ha programado la *Lista_1 (trailers)*, la cual se ha definido para la *Ubicación cines_hd*, sobre el *Escenario 0*. Esta la lista de reproducción, *Lista_1 (trailers)*, está formada por dos archivos de vídeo de calidad y características muy diferentes como se muestra en la Tabla 20.

A continuación, se muestra el resultado de las pruebas iniciales en la situación en la que no se ha implementado la arquitectura *Auto-Proxy* y, por tanto, no se hace el uso de la sincronización de los archivos multimedia de modo local en el *ERC*. Esta prueba se ha denominado *01001 (NAP/NS, No Auto-Proxy y No Sincronización)*:

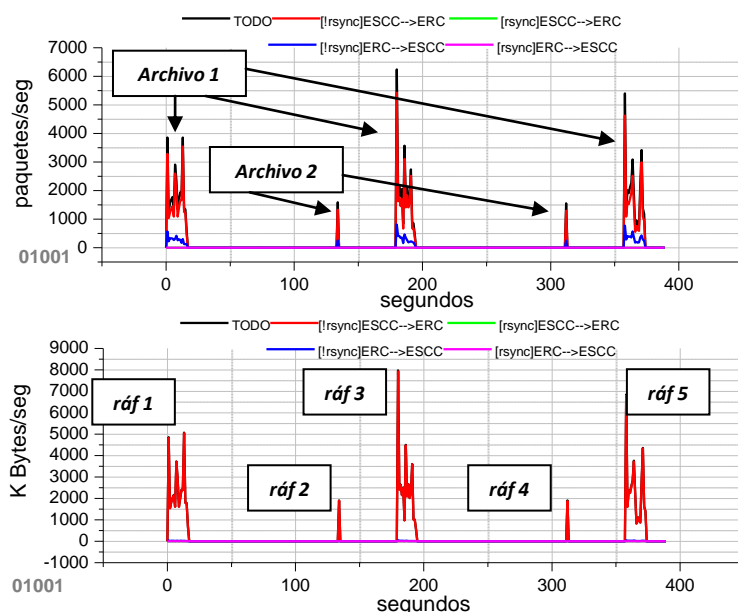


Ilustración 66.-. Prueba 01001. Marco forzado a funcionamiento en No arquitectura Auto-Proxy y No Sincronización de datos. Espacio de Trabajo Simple sobre equipos virtualizados en el mismo equipo anfitrión. Interfaz de red virtualizada en los equipos.

En la ilustración anterior se ha representado tanto los *paquetes/seg*, como los *KBytes/seg* generados por el *Marco* sobre la red *virtualizada* en cada instante. Se puede apreciar cierto tráfico desde el equipo *ERC* hacia el *ESCC* diferente del protocolo *rsync*. Este tráfico corresponde con los *ACKs* (*ACKnowledgement, acuse de recibos*) del protocolo *TCP* en el transporte del contenido multimedia mediante el mecanismo *HTTP Streaming* utilizado.

También, en la Ilustración 66, se puede verificar que el tráfico es debido al patrón de dos ráfagas similares, las cuales se repiten a intervalos constantes en el tiempo. Observando la *lista de reproducción* que se está emitiendo en modo *HTTP Streaming* (ver Tabla 19), se aprecia que secuencialmente se emiten dos archivos de muy distintas características. Si observamos la adquisición de los flujos de red en este modo de funcionamiento, se puede llegar a algunas conclusiones acerca de estos archivos, sin tener información fehaciente sobre ellos:

- El **archivo 1** tiene una duración, según la Ilustración 66, de aproximadamente *02:13 seg* (132 seg), por lo que el cálculo es bastante aproximado al real, ya que el archivo tiene una duración exacta de *02:10 seg* (según la aplicación *VLC*). Se puede calcular el área de la curva (representada en *KBytes/seg*) que cubre la segunda y tercera ráfaga del *archivo 1* mediante la integración de ésta a lo largo del tiempo utilizando aplicaciones informáticas³⁶. Integrando el intervalo, se obtiene:

Ráfaga	Intervalo Tiempo	Área
3 (Archivo 1)	177-194	37937,454101563
5 (Archivo 1)	355-373	37498,087890625

Tabla 25.-. Cálculo del área del Archivo 1 en varios intervalos (ráfaga) de la prueba 01001.

Por tanto, a partir de los datos adquiridos en la red se puede deducir que el archivo ocupa un total de *37,05 MBytes* en la tercera ráfaga (ráf. 3) y un valor de *36,62 MBytes* en la quinta (ráf. 5). Estos valores se aproximan mucho al valor real del

³⁶ En el *proyecto final de carrera* se ha realizado la integración de la curva utilizando la aplicación *OriginPro 8*.

archivo (35,66 MBytes) y son una buena referencia para el cálculo del tamaño del mismo. Así, podría achacarse el resultado mayor y variable debido a las cabeceras de los paquetes intercambiados.

- Para el **archivo 2**, y utilizando la misma técnica o procedimiento que para el *archivo 1*, se obtienen también información importante acerca del archivo que se está reproduciendo. La distancia en segundos con respecto a la siguiente ráfaga son 45,5 seg exactamente por lo que coincide con el valor de la duración del archivo de forma muy precisa (el real son 45 seg). Con respecto al tamaño del archivo se obtiene integrando el área de la curva que muestra el resultado en KBytes/seg:

Ráfaga	Intervalo Tiempo	Área
2 (Archivo 2)	132-135	1887,0224609375
4 (Archivo 2)	309-314	1885,088671875

Tabla 26.-. Cálculo del área del Archivo 2 en varios intervalos (ráfaga) de la prueba 01001.

Con ello se obtiene el tamaño de estos archivos un valor de 1,86MB y 1,84MB, en las respectivas ráfagas, con lo que de nuevo obtenemos un valor también muy aproximado al valor real el cual es de 1,75MB.

Centrándonos en una de las ráfagas mayores, la segunda de ellas (ráf 3 de la Ilustración 66), se puede analizar con algo más de detalle el funcionamiento del *Marco* en esta configuración. En la siguiente ilustración se muestra, de forma conjunta, el resultado obtenido de la adquisición en *paquetes/seg* y en *K Bytes/seg*:

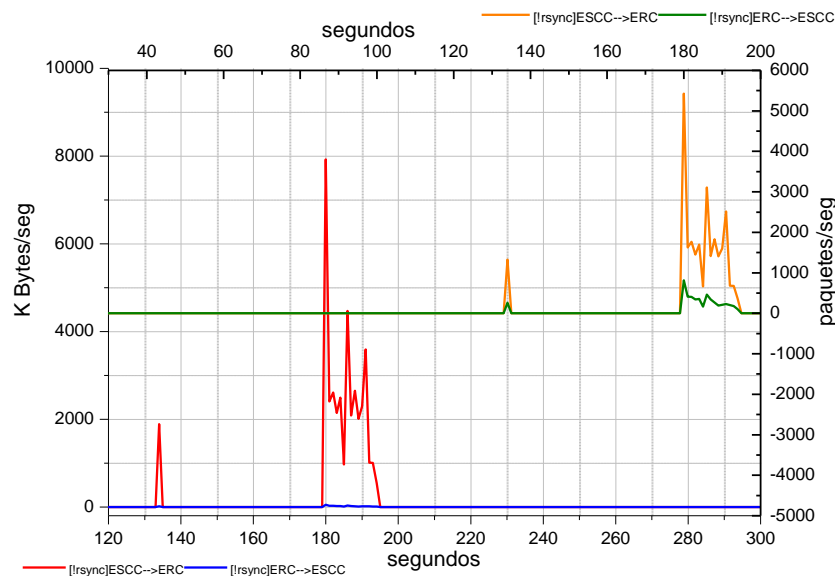


Ilustración 67.-. Detalle de Prueba 01001. Se muestran los paquetes/seg y los K bytes/seg en la misma gráfica

En la Ilustración 67 se muestra; por una parte, en rojo y azul, en la esquina inferior izda. y referenciada sobre el eje de ordenadas izquierdo y de abscisa inferior, el ancho de banda (K bytes/seg) en cada instante ocupado en la prueba 01001; en relación al tráfico generado por el intercambio de archivos bajo demanda (NAP/NS). Por otra parte, en la parte superior derecha, con colores naranja y verde y sobre el eje de ordenadas derecho y abscisa superior, se representa el número de paquetes/seg intercambiados en cada instante.

Por tanto, se representa, en el mismo gráfico, con los ejes de abscisas desplazados, el mismo tráfico pero en distintas unidades. Así, se puede, de forma visual, analizar el comportamiento del sistema. Se puede observar que la forma de las curvas, del tráfico generado por el equipo *ESCC*, establece que existe una proporción en relación al número de paquetes intercambiados con respecto al *ancho de banda* ocupado; sin embargo, el tráfico responsabilidad del equipo *ERC* no establece esta proporción, sino que el número de paquetes enviados es proporcional al número de paquetes enviados por el *ESCC* y por tanto recibidos por el *ERC*. Es decir, son la respuesta a los paquetes recibidos desde el *ESCC* y, por tanto, corresponden a los paquetes *ACKs* del tráfico del protocolo *TCP*. Estos paquetes del tipo *ACK*, los cuales pueden ser numerosos, generan poco tráfico en relación al *ancho de banda* consumido por el sistema y son debidos al particular mecanismo de establecer el *HTTP Streaming* en el *Marco*.

A continuación, en la Ilustración 68, se presenta de forma gráfica y en *K bytes/seg* el resultado de las pruebas obtenidas cuando se activa el mecanismo de *sincronización de contenido* mediante la aplicación *rsync*:

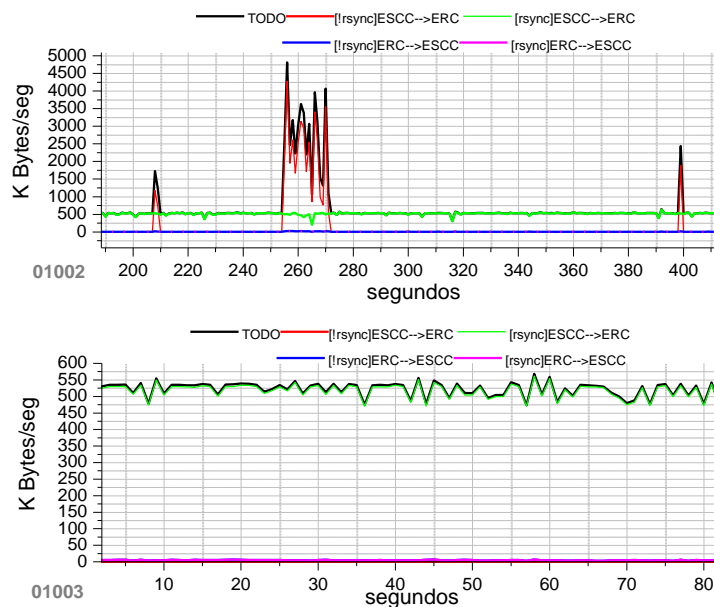


Ilustración 68.-. Pruebas 01002 y 01003. Activación de la sincronización de contenido en el bloque de pruebas 0100X.

El gráfico superior (01002) de la Ilustración 68 representa la peor situación de utilización de *ancho de banda* a la que se ha forzado que funcione el *Marco* (*NAP/CS*) (ver apartado 8.3.2 del Capítulo 8, pág. 216). En las pruebas realizadas se ha forzado esta situación (*NAP/CS*), el cual es un caso bastante particular. Esta situación la podemos encontrar en el funcionamiento del *Marco* con todas las funcionalidades activadas cuando:

- El *ERC* inicia por primera vez. La primera vez que inicia, recupera el tráfico *HTTP Streaming* directamente desde el *ESCC* e inician la sincronización del contenido almacenándolo de forma local en el *ERC* haciendo uso de la aplicación *rsync*.
- El *ERC* está sincronizando el contenido multimedia del servidor y está aplicando *HTTP Streaming* desde el servidor debido a que aún no dispone del contenido almacenado de forma local. Esta situación es indeseable y puede evitarse con la correcta planificación de la configuración; es decir, programar las listas sobre las

ubicaciones después de verificar que el contenido multimedia está almacenado de forma local en el *ERC*.

En esta situación (*NAP/CS*) nos encontramos dos tipos de tráfico:

- Observamos tráfico a ráfagas de duración determinada que aparece periódicamente y que está relacionado con la descarga de contenido directamente desde el *ESCC* a través de la técnica *HTTP Streaming* utilizada. La duración de la descarga (cada ráfaga) depende del tamaño del archivo que se pretende reproducir y del *ancho de banda* disponible en la red. La periodicidad corresponde a que la *lista de reproducción (Lista_1: trailers)* configurada para la *Ubicación (cines_hd)* del *ERC* está formada por dos archivos que van repitiendo su reproducción a modo de bucle (ver Tabla 20), por lo que está determinada por la duración en la reproducción de los vídeos. En la prueba *01002*, el uso del *ancho de banda* es muy elevado y es debido a que no hay electrónica de red y la comunicación se establece mediante la interfaz de red *virtual* creada (*vmnet1*) en el *equipo anfitrión*.
- El otro tipo de tráfico que aparece más o menos constante (entorno a *525KBytes/seg*), se debe a la aplicación que realiza la *sincronización del contenido* de forma local en el *ERC*. Es decir, es debido a la sincronización mediante la aplicación *rsync* y el valor establecido para el bloque de pruebas *0100X* se ha fijado a *512KBytes/seg* (ver Tabla 22). La configuración se realiza a través de la configuración de la *interfaz gráfica de usuario* programada para el administrador (ver APÉNDICE C.II Manual del Usuario Administrador del *ERC*, CONFIGURACION >> Ancho de Banda Sincro).

El gráfico inferior (*01003*) representa la situación del *Marco* en la que el sistema está actualizando contenido y se está aplicando *Auto-Proxy* sobre los archivos que se están reproduciendo en el *ERC* mediante el uso de un *SPC*. Es decir, esta situación se puede producir, en el *Marco* desarrollado, cuando se sube un nuevo archivo al servidor (y, por tanto, aún no está disponible en el *ERC*) y además este archivo está ausente en la configuración de la *lista de reproducción* de la *Ubicación* del *ERC* evaluado.

Se aprecia claramente sobre las gráficas la gran diferencia del *ancho de banda* utilizado por las ráfagas en las que se reproduce el archivo a través de *HTTP Streaming* con respecto a la situación en la que sólo hay tráfico de sincronización de contenido. Esto es debido a que la aplicación, intenta hacer uso de todo el *ancho de banda* disponible para la descarga progresiva del archivo. Se observa que el mecanismo de limitación del *ancho de banda* desde el *ERC* funciona de forma apropiada en ambos casos ya que se limita el *ancho de banda* a *512 Kbyte/s* en las pruebas *01002* y *01003* (ver Tabla 22). Se observan pequeñas oscilaciones en el uso del *ancho de banda* por el *ESCC*, las cuales son debidas a la correcciones realizadas por el aplicación (y protocolo) *rsync*, con el objetivo de mantener la media de utilización de *ancho de banda* configurado.

Con el objetivo de establecer una confirmación del comportamiento del *Marco* observamos en detalle el funcionamiento en la peor situación posible; es decir, en la configuración *01002*, se obtiene la siguiente gráfica:

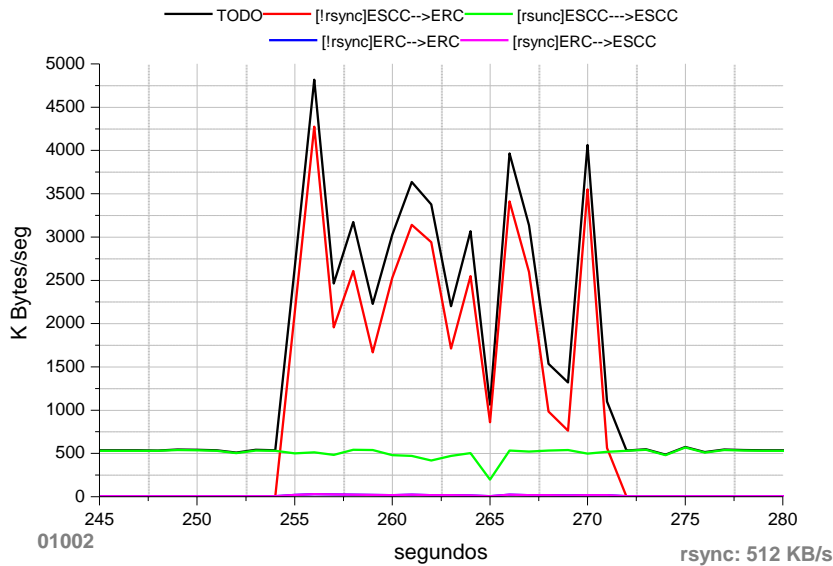


Ilustración 69.-. Detalle prueba 01002

Se observa en la gráfica de la prueba 01002, la cual muestra el *ancho de banda* utilizado por cada equipo en función del tipo de tráfico más representativo, varios máximos y mínimos locales de utilización del *ancho de banda* en la ráfaga *HTTP Streaming*, produciéndose un rizado, más o menos abrupto, en el envío del archivo a reproducir desde el *ESCC* al *ERC*.

Estos rizados están relacionados con que el sistema receptor está recuperando el flujo mediante *HTTP Streaming* y el uso de *TCP*. Según va recibiendo los paquetes los va almacenado en un *buffer* para su reproducción. El receptor, para la reproducción del contenido, según la capacidad de la red, puede llenar el *buffer* completamente, por lo que la aplicación, a nivel de transporte a través del protocolo *TCP*, establecerá un control de flujo de acuerdo a según se vaya descargando el *buffer*. De esta forma, si el flujo de reproducción es menor que la capacidad de la red, se establecerá un rizado conforme a la descarga del *buffer* y al control de flujo establecido. Adicionalmente, el hecho de que el rizado sea tan abrupto, podría estar relacionado con el uso de la *CPU* compartida por los diferentes *equipos virtuales* y el *equipo anfitrión*. Es posible que la aplicación que reproduce el contenido en el *ERC* no sea capaz de atender la reproducción debido a que no se puedan satisfacer las necesidades de *CPU*, por lo que no pueda descargar el *buffer*. Esto hará que el *buffer* se llene aplicando control de flujo a nivel *TCP*. Además, es posible que las necesidades hagan que la *CPU* esté desbordada para el proceso de generación de paquetes lo que contribuiría a un rizado mayor. Estas ideas se refuerzan al verificar cualitativamente que la reproducción de los videos se ralentiza cuando se establecen las pruebas con un único equipo anfitrión lo que hace que la descarga de los *buffer* necesariamente sea más lenta en el *ERC*.

Como se ha adelantado, el principal problema de la emulación del *Marco* con esta estrategia es el uso constante de la *CPU* de la máquina anfitriona. Al virtualizar varios equipos y ejecutar la aplicación de adquisición de datos de la red (*Wireshark*) se establece un gran flujo de ejecución en el propio *equipo anfitrión* que repercute en el correcto funcionamiento del sistema. El rendimiento del *Marco* permanece degradado en toda la ejecución, ofreciendo por momentos una difusión ralentizada de los vídeos incluso en modo de funcionamiento de arquitectura *Auto-Proxy*. Esta ralentización en la reproducción de los videos, no está relacionada con la disponibilidad de *ancho de banda*, ya que se obtienen picos de

4,65 MBytes/seg (37 Mbps) en el envío de los videos, por lo que supera con creces las necesidades mínimas establecidas.

Por estas razones, se decide evaluar de forma inicial el sistema con dos equipos anfitriones conectado mediante electrónica de red que posibilita realizar pruebas más sofisticadas. Así, se permite también evaluar el sistema de forma más relajada en aspectos relacionado con la CPU de los equipos y la posibilidad de evaluar el sistema variando aspectos relacionados con el ancho de banda.

9.1.2 *Espacio de Trabajo Simple con Conexión Física. 0200X*

Esta prueba se ha realizado como paso intermedio antes de evaluar el sistema en una situación de *Espacio de Trabajo Distribuido*. La razón de evaluar el *Marco* en esta configuración es debido a que el sistema, al ejecutar diversas aplicaciones (*Wireshark*, *VMware*) y equipos virtualizados (*ERC*, *ESCC*) en un mismo *equipo anfitrión* de forma simultánea, produce que el funcionamiento del *Marco* se degrade de forma considerable debido al uso compartido de un único procesador. Así, cada equipo virtualizado se ejecuta en diferentes máquinas anfitrionas estableciendo entre ellas *hardware* físico de red.

Aunque en este bloque de pruebas no es estrictamente necesario, se utiliza la conexión *wi-fi* desde la interfaz inalámbrica de uno de los equipos portátiles anfitriones. En concreto, se utilizará la conexión hacia el *AP* conectado al *SWITCH*, para la conexión del equipo virtualizado *ERC* (el esquema de esta configuración física puede consultarse en el apdo. 8.2.4), de modo que la interfaz *Ethernet* de ese mismo *equipo anfitrión* se utilizará para simular el *EACCC*. En el otro equipo anfitrión se utilizará la conexión *Ethernet* para conectar el equipo virtualizado *ESCC* y realizar la captura del tráfico mediante *Wireshark*.

El comportamiento esperado se encuadra dentro de los resultados obtenidos en el bloque de pruebas *0100X*, aunque con unos resultados cualitativos mejorados en la reproducción de vídeos. Al hacer uso de electrónica de red de nivel 2, el *Marco* se aproxima a una situación de producción algo más real, ya que intervienen dispositivos electrónicos en la arquitectura de red. Por esta razón, también nos sirve para cerciorarnos de que la configuración establecida en estos equipos es correcta.

En las pruebas de este apartado se han emulado dos posibles sedes con dos listas de reproducción diferentes:

- La primera de las sedes, denominada *Sede Ráfagas* en la Tabla 27, corresponde a un listado de reproducción de archivos similar al utilizado en el bloque de pruebas *0100X*; es decir, difundirá alternativamente un vídeo de alta calidad (*HQ*) y otro de calidad pobre (*LQ*) de forma alternada (ver archivos de *Lista_1* de la Tabla 20).
- Una segunda sede, denominada *Sede Constante* en la Tabla 27, recibirá una *lista de reproducción* en la que los archivos tendrán una calidad media (*MQ*), de modo que todos utilizarán un *ancho de banda* mínimo necesario similar en modo *HTTP Streaming* (ver archivos de *Lista_3_medium* de la Tabla 20: *Zona A/Escenario 1*).

A continuación (Tabla 27), se muestra una tabla con las restricciones de *ancho de banda* mínimo necesario:

Equipo (sede)	BW Disponible	BW Necesario (rsync+HTTP Streaming)		
Sede Ráfagas (lista_1/Escenario 0)				
ERC (ráfagas)	2.483 K B/s (19,4Mbps)	>	798 KB/s	{512 KB/s (rsync)+286 KB/s}
ESCC (ráfagas)				
Sede Constante (lista_3/Escenario 1)				
ERC (constante)	2.483 K B/s (19,4Mbps)	>	583 KB/s	{512 KB/s (rsync)+72 KB/s:Lista_3_medium: Archivo 2}
ESCC (constante)				

Tabla 27.-. Ancho de banda disponible y necesario en cada sede en grupo de pruebas 0200X.

Para deducir los valores del *BW Necesario* de la Tabla 27, se calcula el *ancho de banda* para la peor situación posible en entorno de las pruebas; es decir, el *ancho de banda* necesario cuando se reproduce el archivo que tiene mayores necesidades más el *ancho de banda* configurado para la sincronización de contenido. El *BW Disponible* de la tabla hace referencia a la conexión de los equipos; es decir, el *ancho de banda* máximo disponible para la conexión del *ERC* con el *ESCC*. Éste es el proporcionado por las tarjetas de red de los equipos anfitriones y el método de conexión y ,por tanto, en este caso es el *ancho de banda* entre el *AP* y la tarjeta inalámbrica que conecta el *ERC*.

Sede Ráfagas

Seguidamente (Ilustración 70), se presentan los resultados de las pruebas cuando se establece una *lista de reproducción a ráfagas* con archivos de diferentes calidades:

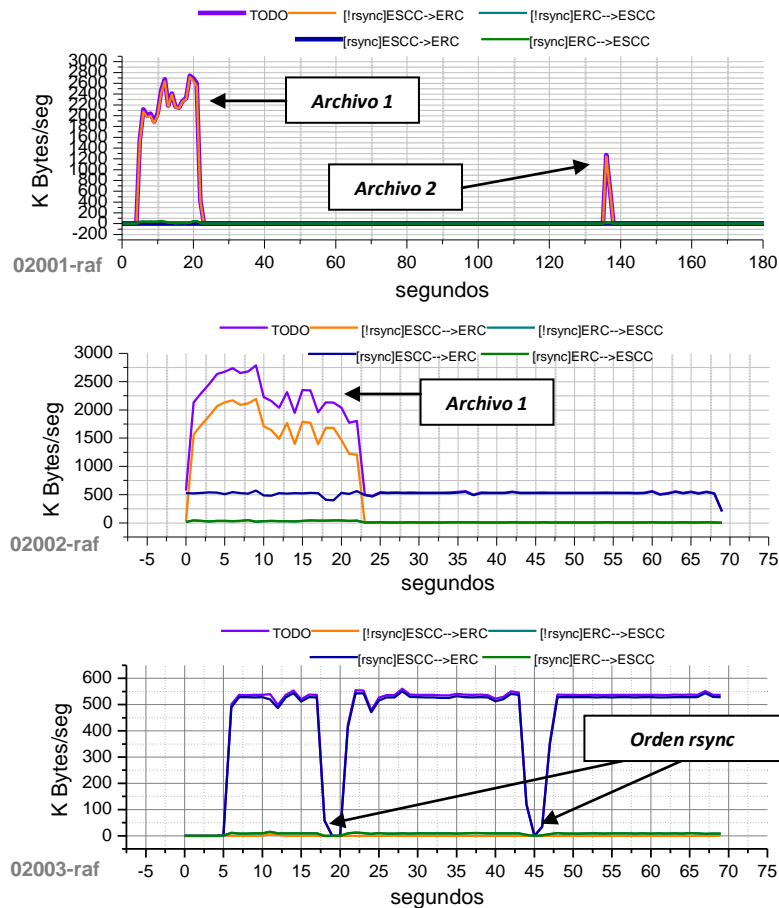


Ilustración 70.-. Sede Ráfagas. El flujo HTTP Streaming de los archivos reproducidos es a ráfagas.

Con la intención de calcular el tamaño del archivo reproducido en el caso de prueba 02002-raf, del mismo modo que se realizó en las pruebas 0100X, se integra el flujo *HTTP Streaming* que se observa en la ilustración relativo al primero de los archivos de la lista de reproducción. Así, el resultado obtenido es el siguiente:

Ráfaga	Intervalo Tiempo	Área
1 (Archivo 1)	0-69	38201,952148438

Tabla 28.-. Cálculo del área del Archivo 1 (Lista_1) en varios intervalos (ráfaga) de la prueba 02001.

Con lo que se obtiene que el archivo intercambiado tenga un tamaño aproximado de 37,3 MB el cual es un valor aproximado al valor real (35,66 MB). Se ha de considerar que al valor obtenido se le ha de restar el tráfico utilizado por las cabeceras de los protocolos; sin embargo, es un valor que se puede considerar una buena aproximación.

Se puede observar también la duración de estos archivos a 133 segundos (02:13 seg) aproximadamente. Por tanto, podríamos calcular también el *ancho de banda* mínimo necesario para reproducir el archivo sin interrupciones, sin más que dividir el tamaño del archivo por los segundos que dura, el cual nos produce un valor aproximado de 287 KB/seg, el cual es un valor también muy aproximado al real (286 KB/s). Se ha de considerar que realizando el cálculo de este modo se asume que el flujo de mínimo necesario para la reproducción del archivo es constante, propiedad esta última que no ocurre en todos los algoritmos de codificación de vídeo.

Se observa también en la última gráfica presentada (02003-raf, en la Ilustración 70), que existen dos nulos en la curva de la última prueba (02003; AP/CS). Estos nulos son debidos a que se aplicó la orden 'RSYNC' en el propio ERC a través de la *interfaz gráfica de usuario* del ERC para forzar la sincronización del contenido desde el ESCC al ERC.

Sede Constante

A continuación (Ilustración 71), se presentan los resultados obtenidos cuando los archivos difundidos mantienen un flujo mínimo de reproducción constante cuando se reproduce un archivo u otro en la *lista de reproducción*:

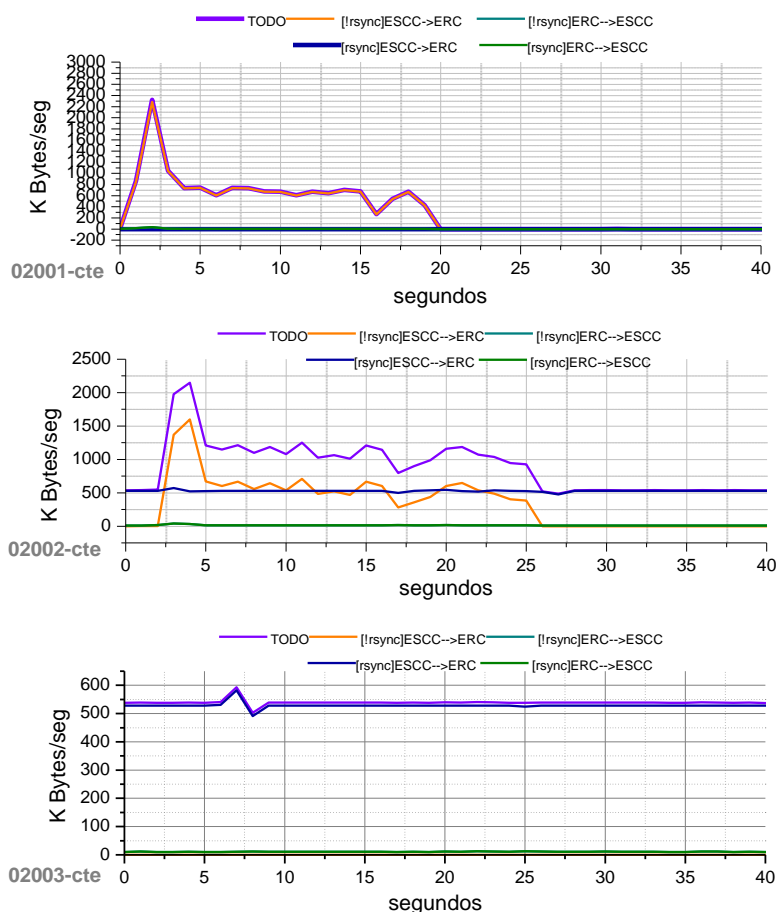


Ilustración 71.-. Sede Constante. El flujo *Pseudo Streaming* necesario de los archivos reproducidos es aproximadamente constante.

La *lista de reproducción de la sede de tráfico constante* establece contenido multimedia cuyos archivos mantienen de forma aproximada la misma calidad de vídeo, estableciendo un flujo constante mínimo necesario. El tráfico que se captura se muestra más suave y constante, esto es debido a las características de los vídeos los cuales descargan el *buffer* más lentamente debido a que el flujo de reproducción necesario es menor.

Se observan algunas similitudes y diferencias entre la *sede a Ráfagas* y la denominada *Constante*.

- 02001. La forma de la curva varía en estas dos situaciones.

- Sede Ráfagas: se observan subidas y bajadas hasta alcanzar un máximo. Son debidas a la aplicación que aplica *HTTP Streaming* que va corrigiendo la cantidad de información que puede recibir para su reproducción. El *buffer* se va vaciando más rápidamente debido a que el flujo en la reproducción es mayor.
 - Sede Constante: se aprecia un flujo alto inicialmente y un flujo menor en el resto de adquisición, suavizándose de modo constante. El mecanismo *HTTP Streaming* ralentiza el flujo requerido del servidor debido a que el *buffer* de la aplicación en el cliente *ERC* se llena y al ser el flujo de reproducción menor, éste se descarga de forma más lenta al ir reproduciendo los vídeos.
- 02002. La forma de las curvas también difieren en la prueba 02002.
 - Sede Ráfagas: la forma de la curva se ve afectada cuando existe otro tráfico presente. Esto es debido a que el *buffer* en el *ERC* tarda más en llenarse.
 - Sede Constante: el tráfico *HTTP Streaming* siguen el mismo patrón que el tráfico *HTTP Streaming* en de la prueba 02001, aunque la descarga del archivo se alarga en el tiempo, debido a que al *ancho de banda* utilizado para la descarga progresiva es algo menor al haber otro tipo de tráfico (*rsync*). Se desprende que parte de este tráfico utilizado es debido al tráfico de sincronización de contenido, el cual es constante según se ha configurado (512 K B/s).
 - 02003. Se observa que el tráfico es contante a lo largo de estas pruebas por lo que verifica que la sincronización se establece al flujo configurado. En la gráfica de la secuencia de la *Sede Ráfagas* se muestran dos nulos que son debidos a aplicar de forma reiterada la orden de sincronización sobre el *ERC*.

La forma de las curvas del tráfico capturado, cuando se establece *HTTP Streaming*, es debida, por tanto, a la velocidad con la que se llena y vacía el *buffer* en el *ERC*. Si hay más tráfico además del correspondiente al *HTTP Streaming*, el *buffer* se llena más lentamente y, si el archivo a reproducir es de mayor calidad, aumenta el flujo necesario para la reproducción de este archivo y, por tanto, el vaciado del *buffer* se realiza más rápidamente debido a que necesita más información para representar cada instante del vídeo.

En aspectos cualitativos observados en la reproducción del contenido multimedia para el bloque de pruebas 02000X, no aparece la ralentización detectada en el bloque de pruebas 0100X. Así, se corrobora que la degradación del *Marco* observada en el bloque de pruebas 0100X, es debida al uso compartido de una sola *CPU* entre varios equipos virtualizados y el equipo anfitrión. Por tanto, el diseño del *Marco* es correcto ya que en el bloque de pruebas 0200X la difusión de los vídeos se establece de forma nítida y sin interrupciones temporales.

9.1.3 *Espacio de Trabajo Distribuido con 1 solo Equipo ERC. 0300X*

En este sub apartado se observa el comportamiento del *Marco* en la situación en la que se introduce una *WAN* en el camino entre el *ESCC* y un *ERC*. Así, en la batería de pruebas 0300X, se limita el *ancho de banda* disponible en la *WAN* entre el *ESCC* y el *ERC*, configurando un valor en el que se permite la reproducción de los archivos de la *lista de reproducción* configurada

(Lista_1) para las *Ubicación* del *ERC*; tanto en modo *HTTP Streaming*, como en modo *HTTP Streaming* con sincronización de archivos simultánea (*NAP/CS*) (peor situación).

En la Tabla 29 se presenta un resumen de la configuración establecida y las necesidades para la *lista de reproducción* configurada (Lista_1; ver Tabla 19 y Tabla 20) en términos de *ancho de banda*:

Equipo	BW WAN (BW Disponible)	BW Necesario (rsync+HTTP Streaming)		
ERC	19,4Mbps 2.483 K B/s (AP)	>>	414 KB/s	{128 KB/s (rsync)+286 Kbps(Lista_1:archivo1)}
ESCC	4000000 bps 488KB/s (serial)	>	414 KB/s	{414 KB/s (ERC)}

Tabla 29.-. Ancho de banda disponible y necesario en cada sede en grupo de pruebas 0300X para cada sede.

El *BW Disponible*, en este bloque de pruebas, queda limitado por el *ancho de banda WAN* del enlace *Serial, Punto a Punto*, entre las sedes del *ERC* y el *ESCC*; ya que éste es menor que el *ancho de banda* que se puede establecer entre el *ERC* y el *SWITCH*, mediante el *AP*. De este modo, en las pruebas se está limitando el *ancho de banda* disponible en el *ERC*, y, para la sincronización de contenido, se ha optado por configurar (128 K Bytes/seg) un valor menor, que en las pruebas anteriores, para así permitir un *ancho de banda* adecuado en la peor situación.

En la Ilustración 72 se muestra el flujo en KBytes/seg intercambiados a lo largo del tiempo, en la situación en la que no se implementa la arquitectura *Auto-Proxy* (03001) y no hay tráfico debido a la sincronización de contenido multimedia en el *ERC*:

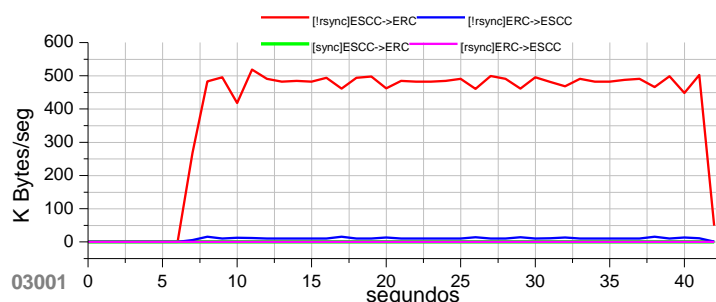


Ilustración 72.-. 03001. Modo de funcionamiento en modo HTTP Streaming.

El tráfico presentado en la Ilustración 72, es el tráfico *HTTP Streaming* generado para reproducir los vídeos que se envía desde el *ESCC* hacia el *ERC*. La ocupación del *ancho de banda* que se obtiene en el *ESCC*, verifica que es en el sentido del *ESCC* hacia el *ERC*; esto es debido a que los vídeos se descargan, como en el resto de casos, progresivamente desde el *ESCC* hacia el *ERC*. La ocupación del *ancho de banda*, según la información mostrada en la Ilustración 72, está en torno a 500 KB/seg. En la prueba 0300X se ha seleccionado un *ancho de banda WAN* de 488 KB/seg entre el *ESCC* y el *ERC* (ver Tabla 22), por lo que los datos recogidos son coherentes con la *WAN* emulada (configuración *Punto a Punto* del enlace *Serial*).

A continuación, se muestra la evolución del sistema en el resto de las situaciones de interés y estables a la que se fuerza el *Marco*:

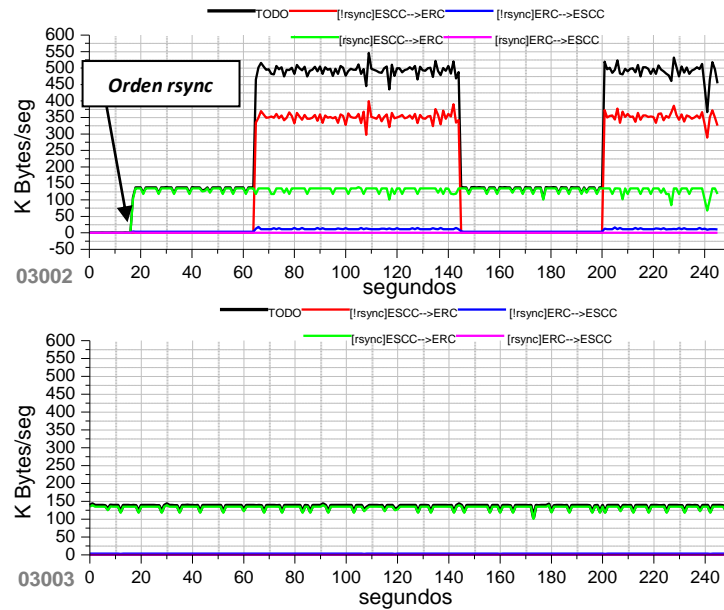


Ilustración 73.-. Resultado de las pruebas 03002 (NAP/CS) y 03003(AP/CS).

Se muestra, en la Ilustración 73, el tráfico debido a las situaciones en las que se fuerza el Marco: NAP/CS (03002) y AP/CS (03003).

El hecho de que aparezca un espacio temporal, en la prueba 03002, sin envío de tráfico *HTTP Streaming*, indica que entre el envío de un archivo y otro ha habido tiempo suficiente para enviar el archivo de forma completa, por lo que se corrobora que el *ancho de banda* de WAN configurado es suficiente para la reproducción en *HTTP Streaming* del archivo. En la prueba 03002, se observa claramente, también, cuándo se proporciona la orden al ERC para que comience la sincronización de contenido: aproximadamente a los 18 seg (*orden rsync*).

En la prueba 03003 se observa que prácticamente todo el tráfico es debido a la sincronización de contenido del ESCC hacia el ERC. Independientemente de que la configuración del *ancho de banda* límite entre el ESCC y el ERC sean 488 Kbyte/s, el uso de *ancho de banda* en esta prueba es el configurado para el tráfico de sincronización de archivos el cual es 128 Kbyte/s.

9.1.4 Espacio de Trabajo Distribuido con 2 Equipos ERC. Configuración límite del BW entre sedes. 0400X

Esta prueba se centra en comprobar el funcionamiento del sistema con más de un equipo ERC funcionando simultáneamente dentro del Marco y forzando al límite del *ancho de banda* necesario en una de las sedes. Así, se ha realizado la misma configuración para cada una de las sedes en aspectos relacionados con el *ancho de banda* y se han programado dos listas de reproducción con necesidades diferentes.

En esta prueba se ha considerado el mismo tráfico establecido en las pruebas anteriores (para la *Ubicación-1*: ERC-1; es decir, la *Lista_1*), más una nueva *lista de reproducción* de *archivos multimedia* difundiéndose para la *Ubicación-2*: ERC-2 (*Lista_2_low*) (ver Tabla 19 y Tabla 20).

En esta prueba también se ha incluido el uso del AP dentro del sistema para conectar un segundo ERC, en la *Ubicación-2* lógica, desde el mismo *equipo anfitrión* que ejecuta el primer

ERC, en la *Ubicación-1* lógica (ver apdo. 8.2.4 Conexión Física mediante AP). De este modo se establece la conexión física del ERC-2 a través de la interfaz *wi-fi* del equipo portátil hacia el SWITCH, aunque se ha de tener en cuenta que esto no repercute en la configuración lógica presentada en el apartado 8.2.3 del Capítulo 8 (Espacio de Trabajo Distribuido).

Seguidamente (Tabla 30), se muestra el *ancho de banda* de cada sede ficticia y el máximo *ancho de banda* necesario por cada *Ubicación*, para reproducir de forma correcta la *lista* completa en la peor situación (*NAP/CS, No Auto-Proxy/Con Sincronización*):

Equipo	BW WAN	BW Necesario (rsync+HTTP Streaming)		
ERC-1	[488(580 normal)-671(ex)] KB/s	\geq	541 KB/s	{256 KB/s (rsync)+286 KB/s (lista_1)}
ERC-2	[488(580 normal)-671(ex)] KB/s	>	293 KB/s	{256 KB/s (rsync)+ 37 KB/s (lista_2_low)}
ESCC	8000000 bps 977 KB/s (serial)	>	834 KB/s	{541KB/s (ERC-1)+293,5KB/s (ERC-2)}

Tabla 30.-. Ancho de banda disponible y necesario en cada sede en grupo de pruebas 0500X para cada sede.

La configuración de la WAN para cada sede se establece en la propia sub interfaz que simula el router de la sede, por lo que el tráfico WAN se modela según la configuración de un acceso *Internet* (como se explica en detalle en el ANEXO B.II Configuración BW WAN en Espacio Distribuido con Varias Sedes) y no sólo a través de la conexión *Serial* (*DTE/DCE*) hacia la sede del ESCC.

La reproducción conjunta de los archivos en la peor situación (*NAP/CS: 04002*) es suficiente para transmitir los archivos de ambas sedes sobre el total del *ancho de banda* disponible en la sede donde se encuentra el ESCC en cualquiera de los casos. Por otra parte, el *ancho de banda* de la Sede 1, *Ubicación-1*, se aproxima al necesario para reproducir el archivo que más *ancho de banda* consume en modo *HTTP Streaming* junto con el tráfico *rsync* configurado para ese equipo. Por tanto, la configuración del *Marco* ha sido tal, que permite un comportamiento correcto en la mayoría de los modos de funcionamiento probados. El *ancho de banda* límite en la sede donde se encuentra el ERC-1 (*Ubicación-1*) tendrá sus consecuencias en el modo de funcionamiento *NAP/CS*, debido a que un archivo en *HTTP Streaming* permite su reproducción para el *ancho de banda* configurado, pero establece la limitación cuando se añade el tráfico de sincronización de archivos (*rsync*).

A continuación (Ilustración 74), se muestra los resultados de las pruebas más significativas:

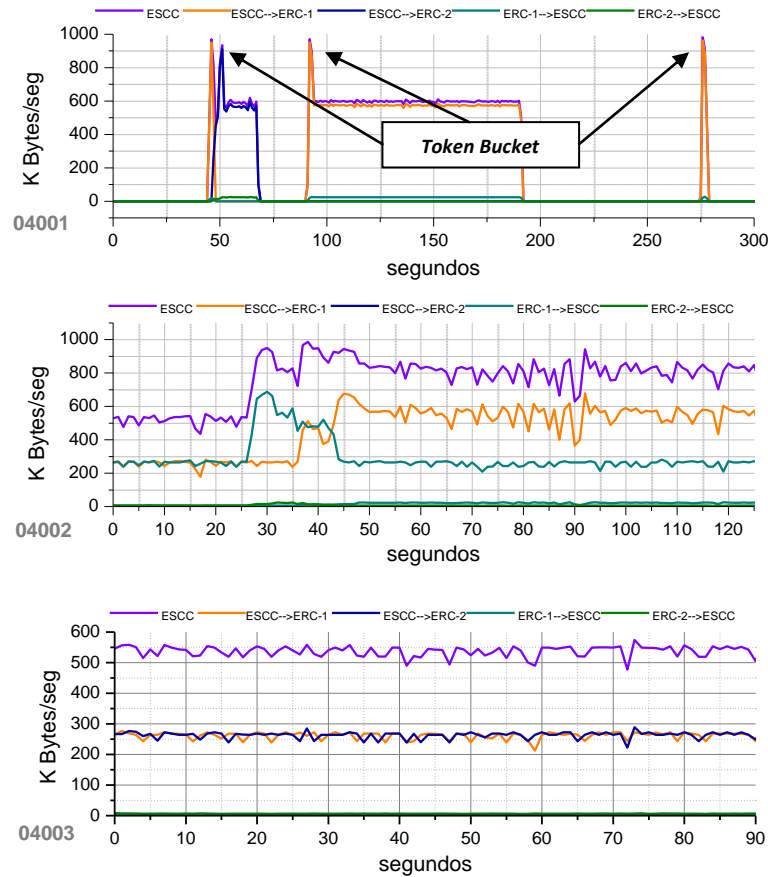


Ilustración 74.-. Presentación de resultados de la prueba 0400[123]. Se presenta el intercambio de tráfico entre equipos.

En la gráfica superior (04001) de la Ilustración 74 se pueden observar explícitamente las ráfagas permitidas por el algoritmo *Token Bucket* aplicado a las sub-interfaces de cada sede (ver APÉNDICE B.II, pág. 303). Se observa que el tráfico se adapta a la configuración de la ráfaga de tráfico normal configurada (580 K Bytes/seg), por lo que se puede considerar que la configuración es apropiada para simular el hipotético *ancho de banda* contratado desde una sede remota.

Como se puede apreciar en la gráfica central (prueba 04002) de la Ilustración 74, se observa aparentemente un comportamiento extraño en la curva que muestra el tráfico enviado desde el *ESCC* al *ERC-1*, atendiendo a las características de los archivos difundidos en la pruebas. Para analizar en detalle el porqué del comportamiento de esta curva, realizamos una división en la representación del tráfico entre del tráfico *HTTP Streaming* y el tráfico *rsync*:

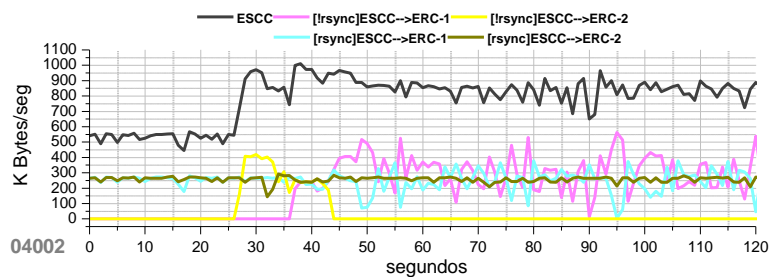


Ilustración 75.-. Detalle de tráfico en 04002 separado por tráfico rsync y no rsync (!rsync) para cada ERC.

En la Ilustración 75 se observa, cuando está presente tanto el tráfico debido a la reproducción de los archivos en modo *HTTP Streaming* como el debido al tráfico *rsync*, que se producen fuertes oscilaciones en las curvas del tráfico capturado en la sede donde se encuentra el *ERC-1*. Estas oscilaciones son debidas a que se está produciendo una ‘disputa’ por el *ancho de banda* disponible entre estos tipos de tráfico (control de congestión de *TCP*). Como se observa, las oscilaciones comienzan cuando empiezan a transmitirse ambos tipos de tráfico en la sede del *ERC-1*; es decir, se puede observar que aunque las necesidades de *ancho de banda* sean menores para reproducir el archivo, el mecanismo de descarga progresiva utilizado (*HTTP Streaming*), intenta hacer uso de todo el *ancho de banda* disponible si el *buffer* aún no se ha llenado. Además, debido a que el *ancho de banda* total se encuentra en el límite para la reproducción del archivo en modo *HTTP Streaming*, cuando existe también tráfico de sincronización y la descarga necesaria del archivo es elevada, se ha observado que se produce que el *ancho de banda* ocupado por este tipo de tráfico (*HTTP Streaming*) se prolongue en el tiempo y sea más próximo al comienzo de la reproducción del siguiente archivo con respecto a la prueba 04001.

Por otra parte, aparentemente el tráfico de la sede donde se encuentra el *ERC-2* es diferente al representado para la sede donde se encuentra el *ERC-1*. Esto es debido a que, aunque el comportamiento en disputa por el *ancho de banda* es similar al de la sede del *ERC-1*, es capaz de transmitir el archivo en un menor tiempo. Esto se produce debido al tamaño del archivo programado, de modo que aparentemente no aparece reflejada esa situación precisamente por las características del archivo a descargar.

9.1.5 Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN Insuficiente en una sede: 0500X.

En el apartado anterior se ha evaluado el sistema en condiciones de funcionamiento límite, cuando se está realizando simultáneamente sincronización y *HTTP Streaming* en alguna de las sedes (04002). En bloque de pruebas 0500X estableceremos el *ancho de banda* necesario para la reproducción en *HTTP Streaming* como insuficiente en una de las sedes según la configuración del *ancho de banda* WAN y evaluaremos el comportamiento del *Marco*.

Las *listas de reproducción* para cada una de las sedes se mantiene de acuerdo a las configuradas en el bloque de pruebas 0400X, reduciendo el *ancho de banda* permitido para cada una de las sedes, a través de la modificación de los parámetros de configuración del algoritmo *Token Bucket* (ver APENDICE B.II) de las sub-interfaces de cada sedes.

El objetivo del grupo de pruebas de 0500X es, por tanto, el de establecer el comportamiento del sistema cuando existen varios equipos *ERC* y el *ancho de banda* utilizado para la reproducción del contenido multimedia es mayor que el disponible en la conexión desde al menos un *ERC* con el *ESCC*. Esta situación se simulará para que se produzca en las sedes, pero sin superar, en suma, el *ancho de banda* total de la sede principal del *ESCC*. Con esta prueba se puede determinar cómo afecta la configuración errónea del *ancho de banda* de una sede de forma concreta y cómo afecta al *Marco* en su conjunto.

Equipo	BW WAN (Disponible)	BW Necesario (rsync+HTTP Streaming)		
		Resultado		
ERC-1	[128(152 normal)-176(ex)] KB/s	<<	541 KB/s	{256 KB/s (rsync)+286 KB/s(lista_1)}
ERC-2	[256(304 normal)-352(ex)] KB/s	>≈	293KB/s	{256 KB/s (rsync)+ 37 KB/s(lista_2_low)}
ESCC	8000000 bps 977 KB/s (serial)	>	834 KB/s	{541KB/s (ERC-1)+293,5KB/s (ERC-2)}

Tabla 31.-. Ancho de banda disponible y necesario en cada sede en grupo de pruebas 0500X para cada sede.

En la Ilustración 76 se presenta el resultado gráfico del tráfico en la sede principal (donde se encuentra el ESCC) debido a cada una de las sedes:

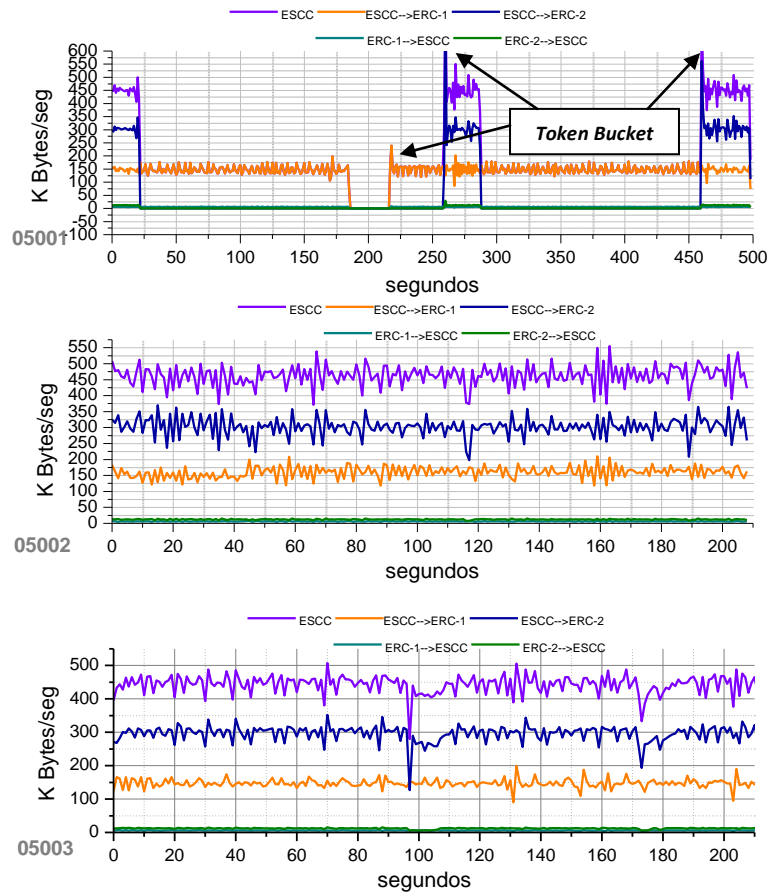


Ilustración 76.- Resultado de las pruebas 0500[123]. Anchos de banda utilizados a lo largo de la prueba.

Como se puede apreciar en la Ilustración 76, el tráfico correspondiente a la sede *ERC-1* sufre diferentes fluctuaciones en todas las gráficas. Con carácter subjetivo indicar que los vídeos de alta calidad emitidos en el *ERC-1*, en las pruebas 05001 y 05002, se entrecortaban y daban apariencia de ralentización de forma constante. Estos errores en la difusión y reproducción de los vídeos no aparecían en las pruebas 05003 y 05004, aspecto que muestra como acertado el diseño de la arquitectura funcional *Auto-Proxy*.

En la gráfica superior de la Ilustración 76 se muestra el tráfico capturado para la prueba 05001. Se observa que cuando se está reproduciendo el *archivo 1* de la *Lista_1* en el *ERC-1*, existe constantemente tráfico *HTTP Streaming*. El hecho de que haya intervalos temporales en los que no aparece tráfico *HTTP Streaming* es debido a que en esos intervalos se está reproduciendo el *archivo 2* de la *Lista_1*, el cual es de mucha menor calidad (ver Tabla 20) y puede ser reproducido de forma correcta para el *ancho de banda* configurado. Se observa que el tráfico (05001) utilizado está en torno a 150 KBytes/seg (configuración de tráfico normal de 152 KBytes/seg, ver Tabla 29). El *archivo 1* reproducido tiene un tamaño real de 35,6 MBytes, por lo que para transmitir el archivo de forma completa del ESCC al ERC, se necesitan 243 segundos (04:03 seg) para el *ancho de banda* configurado. Debido a que la reproducción de archivo dura exactamente 2 min y 10 seg, se evidencia claramente que no es posible reproducir el archivo para el *ancho de banda* configurado de esa sede en modo *HTTP Streaming* y de ahí la ralentización en la reproducción del video. Por otra parte, debido a que el *archivo 2* ocupa en disco tan sólo 1,76 MBytes, se necesitan 12 segundos para transmitir el

archivo completo y, como este tiene una duración de *45 segundos*, se tiene que al menos se tendrán *33 segundos* sin tráfico *HTTP Streaming*. Si se observa en detalle la gráfica para la prueba *05001* (Ilustración 76), se tiene que desde el segundo 185 al 215 (*30 segundos*), no existe tráfico *HTTP Streaming*, por lo que el cálculo obtenido es muy aproximado a lo que cabría esperar (la diferencia empírica se encuentra en el tráfico introducido por las cabeceras de los paquetes; además, para el cálculo exacto también puede influir la configuración del algoritmo *Token Bucket* en la sub-interfaz de la sede).

El tráfico de red para las pruebas *05002* y *05003*, representado en los gráficos mostrados en la Ilustración 76, son aparentemente muy similares; sin embargo, el origen del tráfico difiere en cada tipo de prueba. Para el tráfico presentado para las pruebas *05003* todo el tráfico corresponde a la aplicación *rsync*. En la siguiente ilustración se muestra en detalle la situación en la que, además del tráfico *HTTP Streaming*, se establece tráfico debido a la aplicación *rsync* para la prueba *05002* (NAP/CS):

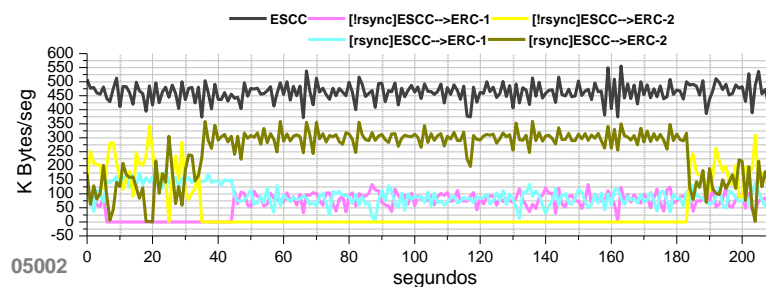


Ilustración 77.-. Tráfico de la prueba *05002* desglosado en función del tipo de tráfico entre la sede central y el resto de sedes.

Según el tráfico de las sedes emuladas para la prueba *05002*, se observa:

- Sede *Ubicación 1 (ERC-1)*: En la sede en la que se reproducen los archivos a ráfagas (*Lista_1*), hay un intervalo al inicio de la secuencia capturada en el que sólo hay tráfico debido al tráfico de la aplicación *rsync*, el cual coincide en un intervalo donde se está reproduciendo el *archivo 2*. Posteriormente existe tráfico tanto *HTTP Streaming* como de sincronización de contenido y, en ambos casos, se producen oscilaciones debido a que las aplicaciones están ‘compitiendo’ por el *ancho de banda*. Este rizado, por tanto, aparece debido a que no hay ningún mecanismo de *conformado de tráfico* o *calidad de servicio* para priorizar el tráfico de una aplicación u otra. Como se observa la duración de la reproducción es mayor que el archivo de mayor calidad.
- Sede *Ubicación-2 (ERC-2)*: En esta sede (*Lista_2_low*) la gráfica aparentemente tiene el mismo patrón que para la sede donde se encuentra el *ERC-1*; sin embargo, el tráfico necesario para reproducir los archivos en modo *HTTP Streaming* permite poder reproducir el archivo acorde con el *ancho de banda* configurado. Así, una vez se ha transmitido el archivo para *HTTP Streaming*, sólo aparece tráfico debido a la sincronización de archivos. Al inicio y al final de la secuencia se observa que existe fuertes oscilaciones debido a ambos tipos de tráfico presentes en la red; no obstante, no se producen los mismos problemas que en la sede configurada en la *Ubicación-1* ya que la calidad de los archivos y el *ancho de banda* disponible, permite reproducirlos de forma adecuada mediante descarga progresiva con *HTTP Streaming*.

9.1.6 Espacio de Trabajo Distribuido con 2 Equipos ERC. Variaciones de ancho de banda WAN necesario en la sede del ESCC. 0600X.

En este apartado se pretende emular una situación de trabajo en la que el *ancho de banda* de la sede principal (donde se encuentra el *ESCC*), con respecto al conjunto del resto de sedes, sea menor que el necesario en varias de las situaciones en el bloque de pruebas. Así, al igual que en el resto de bloques de pruebas, forzaremos al *Marco* para que emule diversas situaciones: *NAP/NS*, *NAP/CS*, *AP/CS* y *AP/NS* (ver apdo. 8.3 Tipos de Pruebas Realizadas del Capítulo 8).

En la prueba se mantienen la *listas de reproducción* con respecto a las pruebas *0400X* y *0500X* y se modifica la configuración de disponibilidad de *ancho de banda* WAN en la sede central donde se encuentra el *ESCC*. Se asigna a cada sede el mismo *ancho de banda* (512 KBytes/seg), de modo que aumenta en la sede donde se encuentra el *ERC-1* con respecto al bloque de pruebas *0500X*.

A continuación (Tabla 32), se presenta la configuración de los *anchos de banda* establecidos en cada sede y las necesidades establecidas por los archivos difundidos en cada *lista de reproducción* configurada:

Equipo	BW WAN	BW Necesario (rsync+HTTP Streaming) Resultado		
ERC-1	[512 (608 normal)-704 (ex)] KB/s	>	414 KB/s	{128 KB/s (rsync)+286 KB/s (lista_1)}
ERC-2	[512 (608 normal)-704(ex)] KB/s	>	165 KB/s	{128 KB/s (rsync)+ 37 KB/s (lista_2_low)}
ESCC	122 KB/s	<	579 KB/s	{541KB/s (ERC-1)+293,5KB/s (ERC-2)}

Tabla 32.- Ancho de banda disponible y necesario en cada sede en grupo de pruebas 0600X para cada sede.

Los datos de la captura en red se han obtenido en la sede principal donde se encuentra el *ESCC* y no en los extremos de los equipos *ERC* (sedes). De los parámetros de configuración de la Tabla 32, se observa que las necesidades de la sede central donde se encuentra el *ESCC* son bastante inferiores a las necesidades de la suma de las sedes.

En la siguiente ilustración, se muestran los resultados obtenidos al emular las dos sedes:

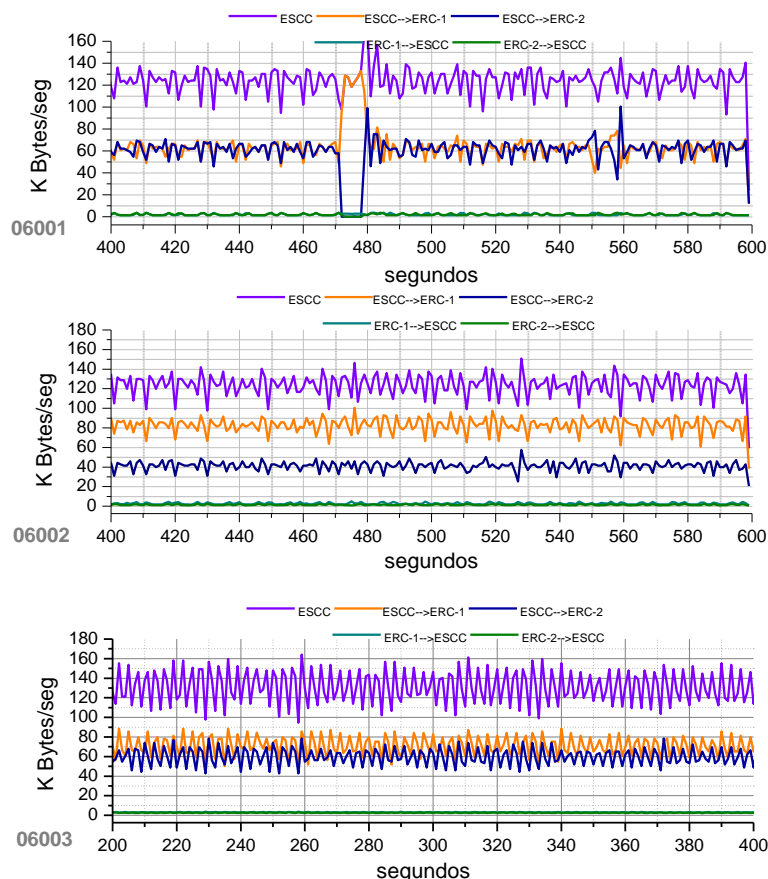


Ilustración 78.- Resultado de las pruebas 0600[123]. Anchos de banda utilizados a lo largo de la prueba.
Ancho de banda limitado por enlace Serial de los routers

Se vuelven a obtener curvas similares; sin embargo, el rizado obtenido es diferente al del resto de pruebas. Se ha de notar inicialmente, que el rizado es muy simétrico, y se verifica que en general el *ancho de banda* se divide de forma simétrica entre las dos sedes. La configuración del *entorno de pruebas* hace que el tráfico se adapte, por tanto, al *ancho de banda* más restrictivo, el cual es el enlace *Serial* (ver Ilustración 62 del Apdo. 8.2.3 Espacio de Trabajo Distribuido).

Se aprecia una discontinuidad para la prueba 06001, en torno al segundo 470 de la secuencia. Esta discontinuidad es debida a que desde la sede donde se encuentra el ERC-2 no hay descarga progresiva del archivo en ese intervalo, lo que produce que todo el *ancho de banda* disponible de la WAN sea ocupado por el tráfico de la sede ERC-1. Es decir, en la sede donde se encuentra ERC-2, se emite en ese intervalo un archivo que necesita un *ancho de banda* relativamente bajo para su reproducción (el archivo de menores necesidades de la lista de reproducción, *Lista_2_low*, son 37 KBytes/seg o 296Kbps; ver Tabla 20). Como el *ancho de banda* disponible para la sede (61 KB/s) permite la descarga progresiva de todo el archivo con anterioridad al final de su reproducción, se establece un intervalo en la que no hay necesidades de *ancho de banda*.

De la Ilustración 78 se puede observar que en la prueba 06002 (NAP/CS) se divide el *ancho de banda* disponible de forma asimétrica para cada sede. Para comprender qué está ocurriendo en este bloque de pruebas representamos el tráfico según sea su origen y tipo:

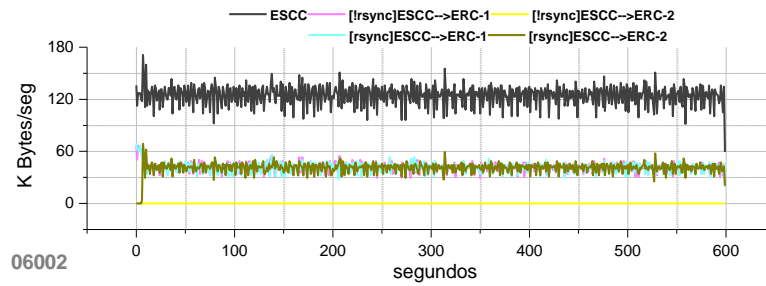


Ilustración 79.-. Detalle del tráfico correspondiente (sin HTTP Streaming en ERC-2)

Se verifica que no existe tráfico relativo al flujo *HTTP Streaming* de la descarga progresiva de los archivos en la sede donde está configurada la *Ubicación-2 (ERC-2)*. Esto es debido a que los archivos multimedia programados en la lista de reproducción (*Lista_2*) se han descargado ya en el propio equipo debido al mecanismo *Auto-Proxy*. No obstante, el *ERC-2* sigue descargando otros archivos a través de la aplicación *rsync*, de modo que el tráfico para esta sede es el relativo a esta aplicación.

Por tanto, el *ESCC* tiene abierto tres flujos: dos para el *ERC-1*, relativos a la aplicación *rsync* y al flujo *HTTP Streaming* y, un tercer flujo, para la aplicación *rsync* en el *ERC-2*. De esta forma, se establece que el servidor atiende estas peticiones de forma simétrica a cada flujo, por lo que la sede *ERC-1* recibe 2/3 (80 KB/s) y la sede *ERC-2* recibe aproximadamente 1/3 (40 KB/s) del ancho de banda total disponible (122 KB/s).

Como aspecto subjetivo, se ha de indicar que de forma cualitativa en este bloque de pruebas, el funcionamiento del *Marco* se vuelve incorrecto cuando no se aplica la arquitectura *Auto-Proxy*, por lo que ante una situación en el que el ancho de banda de la sede central es menor que la suma de los anchos de banda de todas las sedes es necesario habilitar el mecanismo *Auto-Proxy* en el *Marco*. De otro modo el funcionamiento es anómalo en todas las sedes.

9.2 Prueba de Funcionamiento del Marco para Aplicación de Difusión Selectiva con Flujo Streaming Media Multicast

Como se explicado en el Capítulo 6 se han incluido algunas mejoras al *Marco* para reproducir tráfico *Streaming Media*. En este apartado se muestran algunas capturas cuando se ha comprobado la viabilidad de la estrategia de introducir dichas mejoras.

Se han realizado algunas tareas como la instalación de algunas aplicaciones (*VLC* y *plugin VLC*) y la emisión de contenido en directo a través de una cámara web y directamente mediante la emisión de archivos, con la codificación que se presenta en los siguientes apartados, verificando el funcionamiento del *Marco* en ambas situaciones.

9.2.1 Espacio de Trabajo Distribuido con 1 solo Equipo ERC.

Los vídeos emitidos para realizar la prueba corresponden con la lista de reproducción *Lista_1* (ver Tabla 19 y Tabla 20), los cuales tienen características muy diferentes.

En la pruebas se establece una captura mediante la aplicación *Wireshark* [130] para verificar la difusión de contenido en la red. Un ejemplo del tráfico capturado se muestra en la siguiente ilustración:

The image shows a Wireshark capture of network traffic. The top pane displays a list of packets, all of which are UDP packets from source 192.168.100.23 to destination 239.255.0.7, with source port 2767 and destination port 5004. The bottom pane shows the details of a selected packet (Frame 15), identifying it as an Ethernet II frame, an Internet Protocol (IPv4) packet, and a User Datagram Protocol (UDP) packet. The packet length is 1336 bytes, and the checksum is 0xca51 (validation disabled). The packet data is shown in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Info
12	4.768933	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
13	4.768990	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
14	4.769063	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
15	4.769123	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
16	4.769197	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
17	4.769252	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
18	4.972225	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
19	4.972287	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
20	4.972371	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
21	4.972428	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
22	4.972502	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
23	4.972557	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
24	4.972622	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
25	5.186262	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
26	5.186316	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
27	5.186368	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
28	5.189024	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
29	5.189106	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
30	5.189165	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
31	5.189238	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
32	5.189295	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
33	5.189364	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
34	5.189423	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
35	5.189490	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
36	5.189547	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004
37	5.368553	192.168.100.23	239.255.0.7	UDP	Source port: 2767 Destination port: 5004

Frame 15 (1370 bytes on wire (1370 bytes captured))
 Ethernet II, Src: 00:15:b7:9d:13:69 (00:15:b7:9d:13:69), Dst: 01:00:5e:7f:00:07 (01:00:5e:7f:00:07)
 Internet Protocol, Src: 192.168.100.23 (192.168.100.23), Dst: 239.255.0.7 (239.255.0.7)
 User Datagram Protocol, Src Port: 2767 (2767), Dst Port: 5004 (5004)
 Source port: 2767 (2767)
 Destination port: 5004 (5004)
 Length: 1336
 Checksum: 0xca51 (validation disabled)
 0020 00 07 0a cf 13 8c 05 38 ca 51 80 a1 69 fa 7e df8..Q..1..
 0030 38 cf 4d f0 aa 28 47 00 4b 1a a3 2c 08 b7 72 dc 8.M..(G..K....r.
 0040 08 5b 2f e0 06 2b 4d 86 34 a3 3a 48 07 52 9a .[...+M..4....r.
 0050 98 72 f0 ea 92 aa c2 d1 39 8b b5 1d 53 81 72 d2 .[...+M..4....r.
 0060 f8 7f ab f0 a7 e5 b0 33 93 0c 2e 5a 21 5d f9 5e3.S..Z]..A
 0070

Ilustración 80.-. Tráfico Multicast Capturado de la emisión del Archivo 1 de la Lista_1.

El flujo se establece desde el *equipo anfitrión* mediante la *aplicación VLC* utilizando, entre las alternativas disponibles (*RTP/AVP*, *RTP/MPEG-TS*, *MS-WMSP (MMSH)*, *RTSP*, *Icecast*, *HTTP*), el *protocolo RTP/MPEG-TS (Transport Stream)* para el transporte de los vídeos. Se observa que el flujo de datos se realiza mediante el *protocolo UDP* por lo que se establece una *comunicación Streaming Media*, tal y como se ha definido en el proyecto. En la ilustración anterior se puede verificar que la captura se establece en la *dirección IPv4 Multicast* y el *puerto UDP* que se ha configurado: *239.255.0.7:5004*. La codificación de los vídeos y el audio se ha mantenido por defecto a los parámetros del perfil *H.264 + AAC (TS)* de la *aplicación VLC*.

A continuación (Ilustración 81), se muestran algunas de las capturas de red obtenidas al emitir los archivos de la *Lista_1* en modo *Streaming Media Multicast*, mediante el *protocolo RTP/TS (Transport-Stream)*:

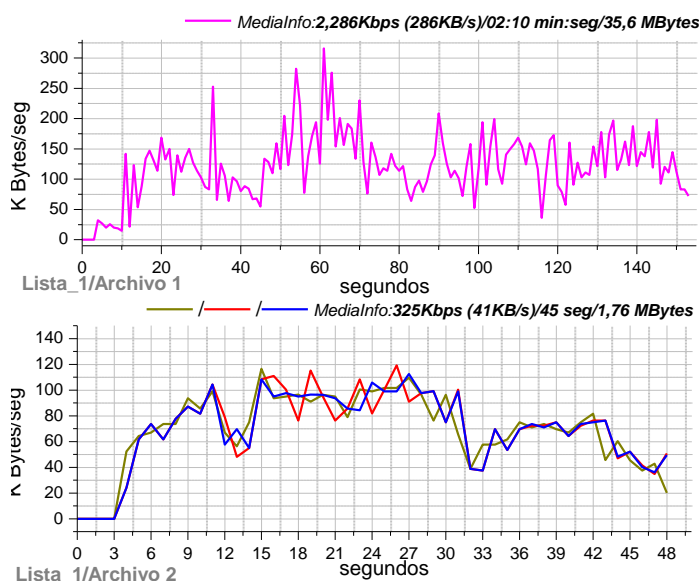


Ilustración 81.-. Capturas de archivos en flujo Streaming Media Multicast para la Lista_1 emitidos desde VLC con el perfil por defecto: H.264 + AAC.

Los valores de los resultados obtenidos para el *archivo 1* y *2*, difieren con respecto a los devueltos por la aplicación *MediaInfo* [131]. Esta diferencia es debido a que la emisión del contenido *Streaming Media* se realiza de acuerdo a los parámetros del perfil de emisión por defecto configurados en la aplicación *VLC*, el cual está referido al perfil *H.264+AAC(MP4)* por lo que el archivo se codifica con otros códec diferentes antes de la emisión.

La importancia de la prueba radica en que es posible reproducir el flujo de red mediante el *plugin* instalado sobre el navegador.

9.2.2 Espacio de Trabajo Distribuido con 1 solo Equipo ERC. Live Streaming

A continuación (Ilustración 82), se muestra una captura de red sobre la emisión en directo de un flujo *Multicast* mediante la emisión de vídeo con una webcam:

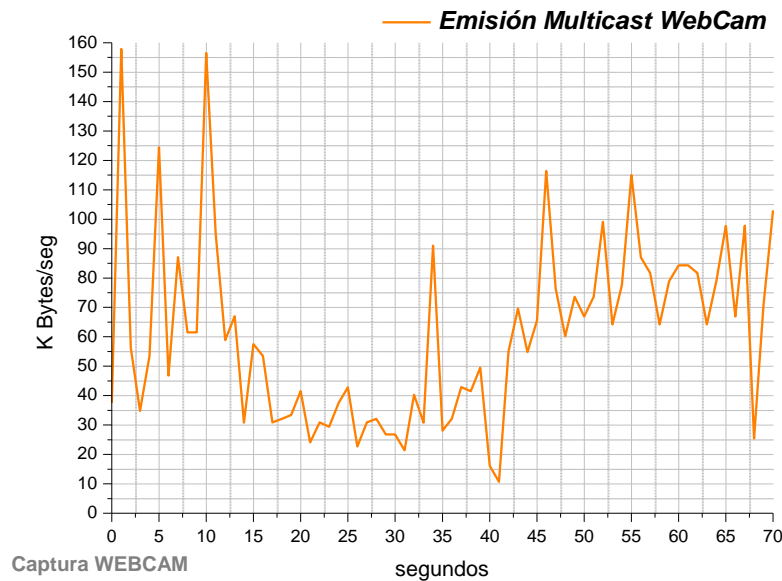


Ilustración 82.-. Captura en red de emisión en directo mediante Webcam y Flujo Multicast.

Se observa que el flujo *Multicast* generado por una *webcam* mediante la aplicación *VLC* establece un flujo considerable sobre la red y éste puede ser diferente atendiendo a la codificación del vídeo. Los parámetros de codificación y emisión han sido los proporcionados por defecto por la aplicación *VLC* en la versión 1.1.7.

Con estas dos pruebas se ha podido mostrar que la estrategia utilizada para mejorar el marco es válida. Así, mediante la utilización de *plugins* de aplicaciones de terceros sobre el navegador es posible utilizar el *Marco* para otras aplicaciones de *Difusión Selectiva*, diferentes a la *Cartelería Digital*, por lo que puede considerarse como una estrategia adecuada para añadir nuevas funcionalidades como es, por ejemplo, la recepción de flujo en directo y la reproducción de flujos *Multicast*.

9.3 Pruebas Transversales

9.3.1 Prueba Transversal: Secuencia de Inicio del ERC. T001

La secuencia de inicio de un *ERC* permite verificar el comportamiento esperado del *Marco*, así como verificar el intercambio de información entre el equipo *ESCC* y el *ERC*. La secuencia se

produce en todos los casos, independientemente de que se esté aplicando arquitectura *Auto-Proxy* o no; por lo que se considera como una prueba transversal.

Así, la secuencia de inicio capturada en cualquiera de las pruebas será similar a la que se muestra en la siguiente ilustración:

No.	Time	Source	Destination	Protocol	Info
18736	34.330991	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-local/repescPHP/repescPHP.php?IP_serv=192.168.103.7&IP_backup=192.168.103.1 W
20744	38.331835	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (text/html)
20800	38.350383	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-local/archivos/IMAG/cine_claqueta.jpg HTTP/1.1
21499	39.640740	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
21946	47.923699	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-as3/reproductorEsc/repZona.swf?escPHP=1&zonaPHP=A&servidorPHP=192.168.103.7&u
21951	47.951901	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-as3/reproductorEsc/controlsocket.swf?servidorPHP=192.168.103.7&servidorSocket
26137	48.308809	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
26238	48.403011	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
26608	49.074944	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-local/repescPHP/reproductorMotorPHP/configEscZonaListaArchivos.php?nocache=0.
27153	50.013172	192.168.103.9	192.168.103.7	HTTP	GET /favIcon.ico HTTP/1.1
27155	50.017346	192.168.103.9	192.168.103.7	TCP	39472 > 843 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=1072612 TSER=0 WS=4
27156	50.017542	192.168.103.7	192.168.103.9	TCP	843 > 39472 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=8030917 TSER=1072612 WS=2
27157	50.017660	192.168.103.9	192.168.103.7	TCP	39472 > 843 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1072612 TSER=8030917
27158	50.019109	192.168.103.9	192.168.103.7	HTTP	GET /crossdomain.xml HTTP/1.1
27376	50.368777	192.168.103.9	192.168.103.7	TCP	39472 > 843 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=23 TSV=1072700 TSER=8030917
27377	50.368976	192.168.103.7	192.168.103.9	TCP	843 > 39472 [ACK] Seq=1 Ack=24 Win=5792 Len=0 TSV=8030952 TSER=1072700
27414	50.428255	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (text/plain)
27433	50.461595	192.168.103.7	192.168.103.9	HTTP/XML	HTTP/1.1 200 OK
27959	51.486463	192.168.103.7	192.168.103.9	TCP	843 > 39472 [PSH, ACK] Seq=1 Ack=24 Win=5792 Len=337 TSV=8031064 TSER=1072700
27960	51.486601	192.168.103.9	192.168.103.7	TCP	39472 > 843 [ACK] Seq=24 Ack=338 Win=6912 Len=0 TSV=1072979 TSER=8031064
27961	51.486838	192.168.103.7	192.168.103.9	TCP	843 > 39472 [FIN, ACK] Seq=338 Ack=24 Win=5792 Len=0 TSV=8031064 TSER=1072979
27962	51.493733	192.168.103.9	192.168.103.7	TCP	39472 > 843 [FIN, ACK] Seq=24 Ack=339 Win=6912 Len=0 TSV=1072981 TSER=8031064
27963	51.496631	192.168.103.7	192.168.103.9	TCP	843 > 39472 [ACK] Seq=339 Ack=25 Win=5792 Len=0 TSV=8031065 TSER=1072981
28022	51.555462	192.168.103.9	192.168.103.7	TCP	34759 > sunproxymdmin [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=1072996 TSER=0 WS=4
28023	51.555615	192.168.103.7	192.168.103.9	TCP	sunproxymdmin > 34759 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=8031071 TSER=1072996
28024	51.555761	192.168.103.9	192.168.103.7	TCP	34759 > sunproxymdmin [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1072997 TSER=8031071
28025	51.610487	192.168.103.7	192.168.103.9	TCP	sunproxymdmin > 34759 [PSH, ACK] Seq=1 Ack=1 Win=5792 Len=149 TSV=8031076 TSER=1072997
28026	51.610619	192.168.103.9	192.168.103.7	TCP	34759 > sunproxymdmin [ACK] Seq=1 Ack=150 Win=6912 Len=0 TSV=1073010 TSER=8031076
28092	51.770880	192.168.103.7	192.168.103.9	HTTP	HTTP/1.1 200 OK (text/html)
28425	52.376468	192.168.103.9	192.168.103.7	HTTP	GET /div-dat/div-local/archivos/VIDEO/Earth_HD.flv HTTP/1.1

Ilustración 83.-. Captura de tráfico http intercambiado en la secuencia de arranque.

Como vemos el intercambio principal de información se realiza a través del protocolo *http*. A continuación, se muestra un diagrama de flujo de interés, extraído de la aplicación *Wireshark*, donde se muestra la secuencia:

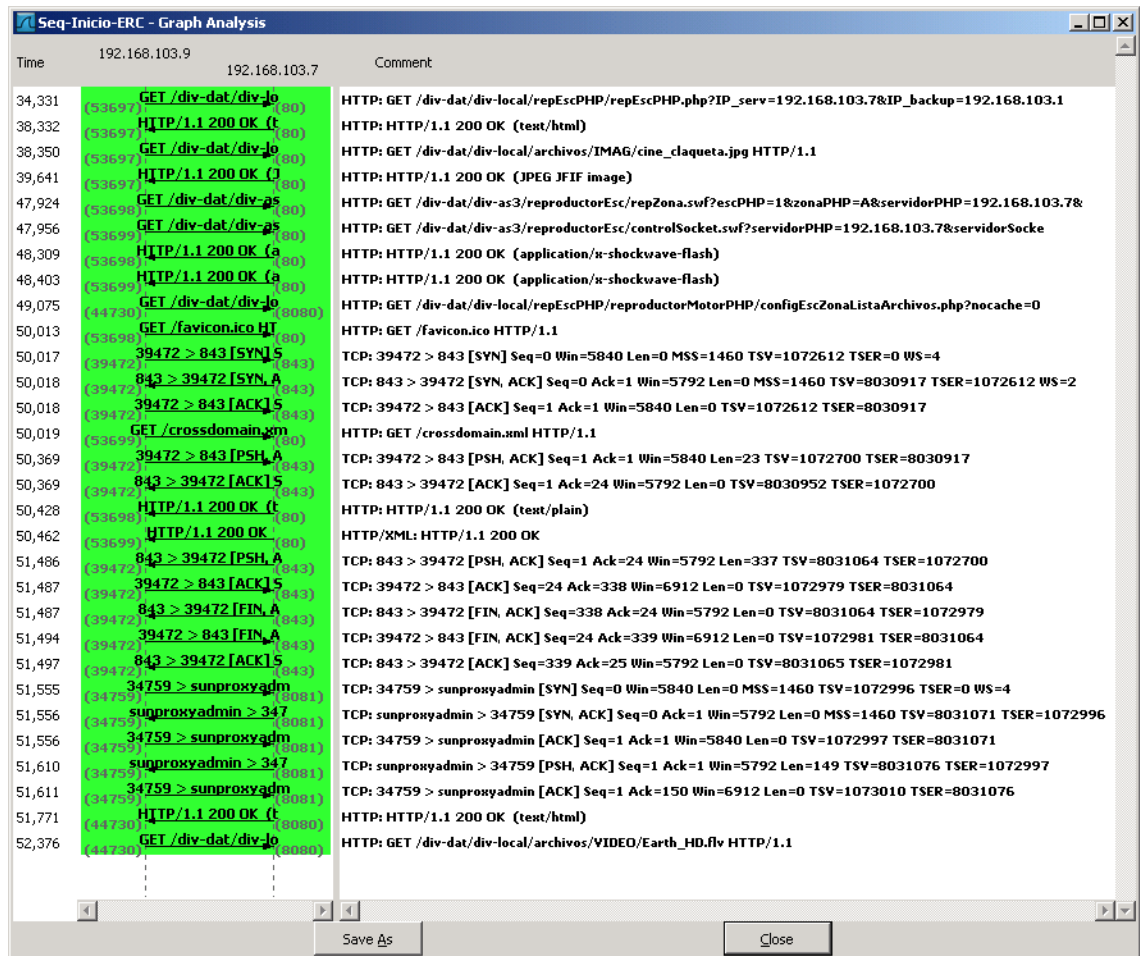


Ilustración 84.- Flujo en la secuencia de arranque del ERC.

La secuencia de inicio será, por tanto, la siguiente:

1. Cuando el *ERC* detecta que el *ESCC* está activo, lo cual lo establece a través de intentos de conexión a la *base de datos* del *ESCC* cada 5 segundos, el *ERC* realiza una petición *http* sobre el *script* (*repEscPHP.php*) que proporciona la página con las *aplicaciones swf* que reproducen el contenido multimedia:

```
192.168.103.9      192.168.103.7      HTTP      GET
/div-dat/div-local/repEscPHP/repEscPHP.php?IP_serv=192.168.103.7&IP_backup=192.168.103.1
HTTP/1.1
```

2. El servidor *ESCC* detecta la dirección *IPv4*, desde la cual se ha realizado la petición *http*, y envía al propio *ERC* la confirmación, junto con una página *html* con la configuración de la *Ubicación* a la que pertenece el equipo.

```
192.168.103.7      192.168.103.9      HTTP
HTTP/1.1 200 OK   (text/html)
```

La cabecera del protocolo *http* de la respuesta, en el ejemplo, es la siguiente:

```
HTTP/1.1 200 OK
Date: Wed, 27 Oct 2010 22:31:15 GMT
Server: Apache/2.2.11 (Unix) DAV/2 mod_ssl/2.2.11 OpenSSL/0.9.8i
PHP/5.2.8 mod_apreq2-20051231/2.6.0 mod_perl/2.0.4 Perl/v5.10.0
X-Powered-By: PHP/5.2.8
Content-Length: 2345
```

```
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Y el contenido, el cual es código html, se muestra a continuación:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml2/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
    />
    <title>Reproductor PHP-SWF</title>

    <SCRIPT language="JavaScript">
    <!--//
      function mensajeJavaScript(mensaje){
        alert('ATENCION: '+mensaje);
      }

      function recargarJavaScript(recargar){

        if(recargar==1){
          window.location.reload();
        }
      }
    //-->
    </SCRIPT>

    <style type="text/css">html, body, div, #zona-A, #zona-B, #zona-C,
    #zona-D, #socket { height: 100%; margin: 0; } </style>
  </head>

  <body>

    <div id='fondo' style='position: absolute; z-index:1; left: 0px;
    top: 0px; width: 100%; height: 100%; visibility: visible;
    overflow: visible; cursor:crosshair; background-color:#ffffff'>

      <img
        id='background'
        src='../archivos/IMAG/cine_claqueta.jpg' alt='' title=''
        style='position:absolute; z-index:0; width:100%;
        height:100%;' />

      <div id='zona-A' style='position: absolute; z-index:
      4;left:25%; top: 25%; width:65%; height: 65%; visibility:
      visible; overflow: visible;background:#000000'>

        <embed
          src='../div-
          as3/reproductorEsc/repZona.swf?escPHP=1&zonaPHP=A&s
          ervidorPHP=192.168.103.7&ubicacionPHP=cineshd'
          quality='high' scale='noscale' salign='lt'
          wmode='transparent' width=100% height=100%
          name='pruebaIMAGCir' align='center'
          allowScriptAccess='sameDomain'
          allowFullScreen='true' type='application/x-
          shockwave-flash'
          pluginspage='http://www.macromedia.com/go/getflashp
          layer' />

      </div >

      <div id='socket' style='position: absolute; z-index: 5;
      left:0%; top: 0%; width:0.01%; height:0.01%; visibility:
      visible; overflow: visible;background:#000000'>

        <embed
          src='../div-
          as3/reproductorEsc/controlSocket.swf?servidorPHP=19
          2.168.103.7&servidorSocketPHP=192.168.103.7&ubicaci
          onPHP=cineshd&IP_equipoPHP=192.168.103.9'
          quality='high' scale='noscale' salign='lt'
          wmode='transparent' width=100% height=100%
          name='controlSocket' align='center'
          allowScriptAccess='sameDomain'
          allowFullScreen='true' type='application/x-
```

```

shockwave-flash'
pluginspage='http://www.macromedia.com/go/getflashp
layer' />
</div >
</div>
</body>
</html>

```

3. Una vez recibido el código `html`, el navegador realiza las peticiones de cada uno de los objetos y elementos que debe contener la configuración de la *Ubicación* como parte de la descarga del protocolo `http` y `html`. El servidor responde con la confirmación y los propios elementos. En el ejemplo presentado son tres los elementos más importantes obtenidos desde *ESCC*:

- i. *Imagen de Fondo de la Ubicación*. Esta imagen no está presente en todos los casos (depende de la configuración).

```

192.168.103.9      192.168.103.7      HTTP  GET
/div-dat/div-local/archivos/IMAG/cine_claqueta.jpg HTTP/1.1

192.168.103.7      192.168.103.9      HTTP  HTTP/1.1 200 OK
(JPEG JFIF image)

```

- ii. *Reproductor de la Zona* (en el ejemplo sólo está la *zona A* configurada en el *Escenario*). Podrían estar configuradas varias *zonas* en el *Escenario* de la *Ubicación* lo que produciría la descarga de más *Reproductores de Zona* (`repZona.swf`).

```

192.168.103.9      192.168.103.7      HTTP  GET
/div-dat/div-
as3/reproductorEsc/repZona.swf?escPHP=1&zonaPHP=A&servidorPHP=192
.168.103.7&ubicacionPHP=cineshd HTTP/1.1

192.168.103.7      192.168.103.9      HTTP  HTTP/1.1 200 OK
(application/x-shockwave-flash)

```

Se ha de notar que en la petición se conoce exactamente la *Ubicación*, el *Escenario* y la *Zona* a la que pertenece el equipo ya que, la página `html` devuelta por el servidor, depende de la dirección *IPV4* desde donde se realiza la petición. De este modo, el *Reproductor de la Zona*, es decir, la aplicación `repZona.swf`, podrá establecer exactamente las peticiones que debe realizar al *ESCC*.

- iii. *Controlador de la conexión socket (puerto 8081) con el servidor* (`controlSocket.swf`). Estará presente en todas las configuraciones y será la aplicación encargada de reconocer las 'órdenes' enviadas desde el *ESCC* a los *ERCs*. Los parámetros recibidos por la aplicación serán similares en los *ERCs* que dependan del *ESCC* que lo proporciona.

```

192.168.103.9      192.168.103.7      HTTP  GET
/div-dat/div-
as3/reproductorEsc/controlSocket.swf?servidorPHP=192.168.103.7&se
rvidorSocketPHP=192.168.103.7&ubicacionPHP=cineshd&IP_equipoPHP=1
92.168.103.9 HTTP/1.1

192.168.103.7      192.168.103.9      HTTP  HTTP/1.1 200 OK
(application/x-shockwave-flash)

```

4. El reproductor de la zona (`repZona.swf`), una vez cargado en el agente de usuario (navegador web), realiza una petición al *ESCC* de la configuración establecida para esa *Zona*, del *Escenario* configurado para esa *Ubicación*. Posteriormente, el *ESCC* responde

con la confirmación y con una página en código *XML* con las listas de reproducción configurada y el listado de los archivos correspondientes.

```
192.168.103.9      192.168.103.7      HTTP   GET
/div-dat/div-
local/repEscPHP/reproductorMotorPHP/configEscZonaListaArchivos.php?n
ocache=0.14408835023641586&esc=1&zona=A&fecha_actual=20101028&modo=p
revisualizar&ubicacion=cineshd HTTP/1.1

192.168.103.7      192.168.103.9      HTTP   HTTP/1.1 200 OK
(text/plain)
```

Se observa que en la petición enviamos la fecha que tiene configurada el *ERC*. Se ha de notar que el envío de esta fecha no es estrictamente necesario, ya que el *ESCC* calcula la fecha (y también la hora) mediante el protocolo *NTP*; sin embargo, es una buena opción incluso para mejoras futuras en la que se podría detectar mediante este método que los equipos no estuvieran sincronizados, pudiendo, incluso, enviar una *lista de reproducción* especial para tales situaciones (*firewalls* mal configurados o pérdidas de servicio *NTP* en el servidor).

Además, se establece la petición del *archivo de política url* para las *aplicaciones swf* con el servidor. Esta petición es necesaria a la hora de implementar la arquitectura *Auto-Proxy*. A través de la petición del archivo *crossdomain.xml*, desde el propio reproductor de la *zona*, se establece la *política*:

```
192.168.103.9      192.168.103.7      HTTP   GET
/crossdomain.xml HTTP/1.1

192.168.103.7      192.168.103.9      HTTP/XML      HTTP/1.1 200 OK
<?xml version="1.0"?>
<!-- http://ESCC/crossdomain.xml -->
<cross-domain-policy>
  <allow-access-from domain="*" to-ports="*" />
</cross-domain-policy>
```

5. La aplicación que monitoriza la conexión *socket* (*controlSocket.swf*) debe obtener el *archivo de política socket*.

La política se obtiene mediante una petición de conexión *TCP*, independiente, y al puerto *843*. De este modo se realiza la petición de los permisos y así, en el servidor, se ha implementado un *servidor socket* que atiende estas peticiones devolviendo la política establecida (en el proyecto se ha optado por una política totalmente permisiva):

```
192.168.103.9      192.168.103.7      TCP
39472 > 843 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=1072612 TSER=0 WS=4

192.168.103.7      192.168.103.9      TCP
843 > 39472 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=8030917
TSER=1072612 WS=2

192.168.103.9      192.168.103.7      TCP
39472 > 843 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1072612 TSER=8030917

192.168.103.9      192.168.103.7      TCP
39472 > 843 [ACK] Seq=24 Ack=338 Win=6912 Len=0 TSV=1072979 TSER=8031064

192.168.103.7      192.168.103.9      TCP
843 > 39472 [FIN, ACK] Seq=338 Ack=24 Win=5792 Len=0 TSV=8031064 TSER=1072979

192.168.103.9      192.168.103.7      TCP
39472 > 843 [FIN, ACK] Seq=24 Ack=339 Win=6912 Len=0 TSV=1072981 TSER=8031064
```

```
192.168.103.7          192.168.103.9          TCP
843 > 39472 [ACK] Seq=339 Ack=25 Win=5792 Len=0 TSV=8031065 TSER=1072981
```

El *ERC*, una vez ha obtenido el *archivo de política de sockets* desde el servidor, se conecta a través de una *conexión socket* al servidor en el puerto escogido para recibir las órdenes del *ESCC*; es decir, al puerto *TCP 8081*:

```
192.168.103.9          192.168.103.7          TCP
34759 > 8081 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=1072996 TSER=0 WS=4

192.168.103.7          192.168.103.9          TCP
8081 > 34759 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=8031071
TSER=1072996 WS=2

192.168.103.9          192.168.103.7          TCP
34759 > 8081 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1072997 TSER=8031071

192.168.103.7          192.168.103.9          TCP
8081 > 34759 [PSH, ACK] Seq=1 Ack=1 Win=5792 Len=149 TSV=8031076 TSER=1072997

192.168.103.9          192.168.103.7          TCP
34759 > 8081 [ACK] Seq=1 Ack=150 Win=6912 Len=0 TSV=1073010 TSER=8031076
```

6. Paralelamente y una vez que se ha establecido el *archivo de política url* para la aplicación *repZona.swf*, se obtiene la lista de reproducción de los archivos para la fecha actual del sistema. Esto se realiza a través de una petición *http* al puerto *TCP 8080*:

```
192.168.103.7          192.168.103.9          HTTP
HTTP/1.1 200 OK (text/html)

<datos_lista  tabla='l_trailersPrueb'  nombre_lista='trailersPrueb'
fecha_lista='20100102' fecha_siguiete='99999999'>
  <valores  select='on'  nombre_archivo='Earth_HD'  extension='flv'
tipo='VIDEO'  ubicacion='loc'  h_inicio='00:00:00'  h_fin='23:00:00'
prioridad='1'  id_multimedia='151'  opcion1='0'  opcion2='5'
texto='Archivo HD!'>6</valores>
  <valores  select='off'  nombre_archivo='anuncioSprite'
extension='flv'  tipo='VIDEO'  ubicacion='loc'  h_inicio='00:00:00'
h_fin='23:00:00'  prioridad='2'  id_multimedia='98'  opcion1='0'
opcion2='5'  texto='Archivo LQ!'>16</valores>
</datos_lista>
```

El resultado es un listado de los archivos con una serie de atributos (los configurados mediante la *interfaz gráfica*) en formato *XML* e información de la fecha para la *lista de reproducción* siguiente. En el ejemplo anterior no existe una lista después de la que se pretende reproducir. La información de la fecha para la siguiente lista es utilizada para publicar un evento en la *aplicación swf* que al producirse realiza la petición de la nueva lista.

7. Finalmente, el *reproductor swf* (*repZona.swf*) evalúa si puede hacer uso de la arquitectura *Auto-Proxy*. Es decir, el *ERC*, a través de la aplicación *swf* evalúa si dispone de los archivos multimedia almacenados de forma local y si no dispusiera de ellos (según el valor del *XML*³⁷ devuelto) enviaría una petición al servidor para obtener así el archivo:

³⁷ Para obtener el valor *XML* de los archivos almacenados de forma local, el *repZona.swf* hace una petición *http* sobre *localhost* (el mismo equipo donde se está ejecutando; es decir el *ERC*). Se utiliza esta estrategia, debido a que estas aplicaciones *swf* (ejecutándose sobre el navegador) no tienen acceso al sistema de archivos del equipo.

```
192.168.103.9      192.168.103.7      HTTP      GET
/div-dat/div-local/archivos/VIDEO/Earth_HD.flv HTTP/1.1
```

En el ejemplo mostrado para la explicación de la secuencia de inicio, la petición del archivo para su reproducción se realiza sobre el *ESCC* debido a que el *ERC* aún no tiene almacenado de forma local el archivo a reproducir estableciendo de este modo *HTTP Streaming* desde el *ESCC*.

9.3.2 Prueba Transversal de Memoria Caché del Navegador. T002.

En esta prueba se pretende configurar la *caché* del navegador con el intento de establecer una arquitectura alternativa a la de *Auto-Proxy* y ver si es posible conseguir resultados de prestaciones similares a ésta. El navegador utilizado (*firefox*), mantiene una estructura de *caché* basada tanto en la memoria *RAM*, como en el disco duro y una tercera para la navegación fueran de conexión (*offline*).

Para la realización de las pruebas se configurará en un *ERC* la *Ubicación* denominada *genUc3m*, en la configuración del *Escenario 2* la cual tiene programada las *Lista_3_high* (ver Tabla 19 y Tabla 20: *genUc3mPruebHQ*) en formato de vídeos de calidad mayor.

Para realizar la prueba se ha de tener en cuenta que los parámetros que controlan la *caché* del navegador pueden ser configurados y consultados. Para modificarlos (incluso crearlos si no se han definido) se ha de escribir '*about: config*' en la barra de navegación de *firefox*. Entre los parámetros de interés para la configuración [132] de la *caché* se tienen:

- **network.http.use-cache:** Cuando se carga una página mediante el protocolo *http*, puede ser almacenada en *caché* por lo que no necesitará ser descargada para ser mostrada de nuevo. Esta opción controla si se recuperan los archivos de *caché* (desde *http* o *https*), ya sea en memoria o en disco. Permite habilitar el *proxy caché* en el propio equipo y de este modo se habilitan la capacidad evaluada en esta prueba. Este parámetro permanece deshabilitado en la aplicación en el resto de pruebas.
- **browser.cache.memory.enable:** Esta opción nos permite determinar si el navegador almacena (*true*) o no (*false*) en la memoria *RAM*.
- **browser.cache.memory.capacity:** Permite establecer la capacidad de la memoria *caché*. Corresponderá con la cantidad máxima de memoria *RAM* (en kilobytes) que usará *firefox* para la memoria *caché*. El valor por defecto es -1 que asigna automáticamente el tamaño de la *caché* dependiendo de la *RAM* física del sistema. Existen unas recomendaciones que aplican, por defecto, según sea la capacidad de la memoria *RAM* del equipo que ejecute el navegador [133]. La configuración no aplica si el parámetro anterior está deshabilitado.
- **browser.cache.disk.enable:** Nos permite establecer el disco duro para almacenar elementos en modo *caché*. Es el parámetro que habilita la configuración para realizar estas pruebas ya que se pretende almacenar el contenido multimedia de gran tamaño.
- **browser.cache.disk.capacity:** Permite seleccionar la cantidad de espacio disco utilizado para la *caché*. El parámetro se configura en *KBytes*.
- **browser.cache.check_doc_frequency:** Este parámetro determina con qué frecuencia se chequea la *caché* para comprobar si existe nueva versión de las páginas que estamos visitando.
 - Si esta preferencia toma el valor 0 *firefox* solo chequea la *caché* una vez por sesión (cada vez que abrimos el navegador).
 - El valor 1 comprueba cada vez que se visita la página.
 - El valor 2 nunca chequea la *cache*.
 - El valor 3, asignado por defecto, comprueba la *caché* en determinados intervalos que son asignados automáticamente.

- **browser.cache.offline.enable** y **browser.cache.offline.capacity**: Estas opciones nos permiten habilitar y limitar el espacio en KBytes de la *caché* utilizada para la navegación offline de Firefox. Los mantenemos deshabilitados en las pruebas.

Con ello, se configura la memoria *caché* del navegador del ERC. Se activa la *caché* en disco con un valor de *2000MB*, como base para establecer el margen para el cual el navegador considerará que los archivos almacenados de forma local puede considerarlos como utilizables y se deshabilitará la *caché* para cuando no se establezca conexión. Así, se ha configurado (*about: config*):

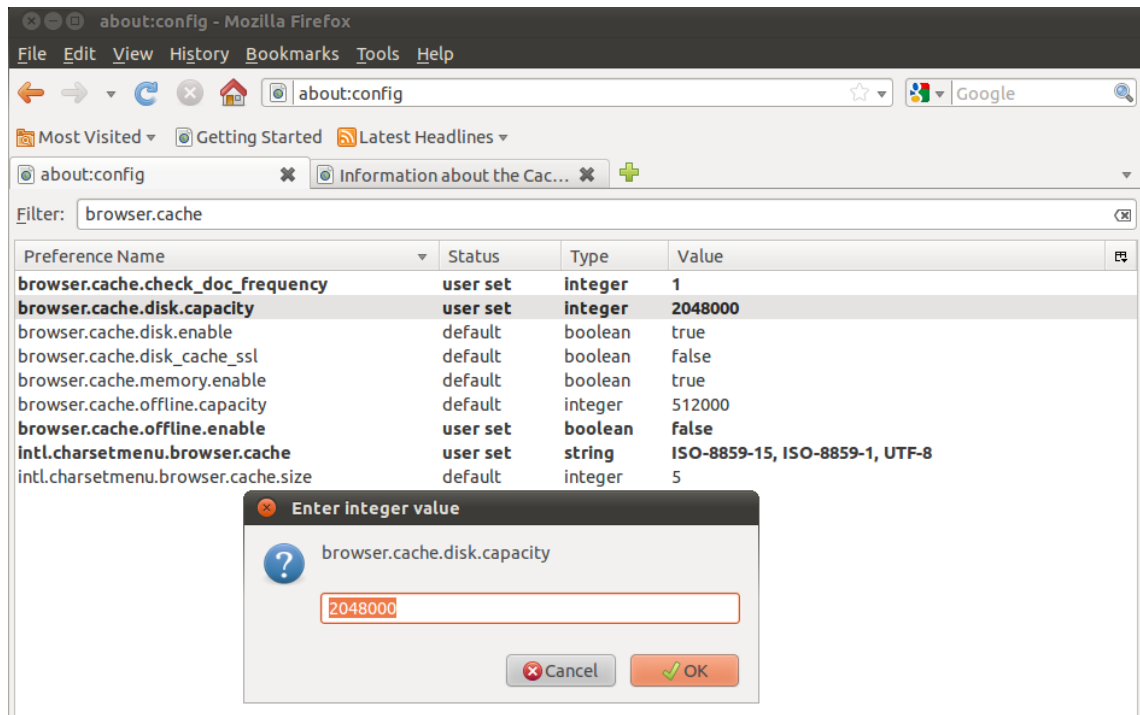


Ilustración 85.-. Configuración de la caché en el navegador firefox para la realización de las pruebas T002.

Estableciendo este margen para los archivos almacenados en la *caché* del navegador, se ejecuta la prueba para aplicar una situación del tipo NAP/NS (*No Auto-Proxy* y *No Sincronización*). Para su evaluación se utiliza el método de previsualización desarrollado para la *interfaz gráfica* de administración, la cual no implementa la arquitectura *Auto-Proxy*. De este modo, se puede evaluar el funcionamiento del *Marco* para ver si es posible obtener un funcionamiento similar al *Auto-Proxy*. Si se obtuviera un resultado similar, se tendría la ventaja de que no sería necesario que se sincronizaran todos los archivos y lo harían sólo los que se deberían reproducir en la *Ubicación*. De esta forma sólo se descargaría una vez cada archivo que se fuera a reproducir.

A continuación, se muestra el resultado al configurar el navegador para hacer uso de la *caché* de acuerdo a los parámetros introducidos:



Memory cache device

Number of entries: 0
Maximum storage size: 16384 KiB
Storage in use: 2101 KiB
Inactive storage: 0 KiB

[List Cache Entries](#)

Disk cache device

Number of entries: 18
Maximum storage size: 2048000 KiB
Storage in use: 228233 KiB
Cache Directory: /home/dtimon/.mozilla/firefox/hk2j2npw.default/Cache

[List Cache Entries](#)

Ilustración 86.-. Estado de la caché del navegador firefox con el sistema en funcionamiento.

El propósito de esta configuración es que los archivos se almacenen en la *caché* del navegador, implementando una arquitectura similar a la *Auto-Proxy* una vez ha reproducido el fichero al menos una vez.

Los archivos de la lista configurada se almacenan de forma local en el (disco) una vez se han reproducido. Así, se puede verificar este comportamiento listando los archivos almacenados en la *caché* del navegador *firefox*:

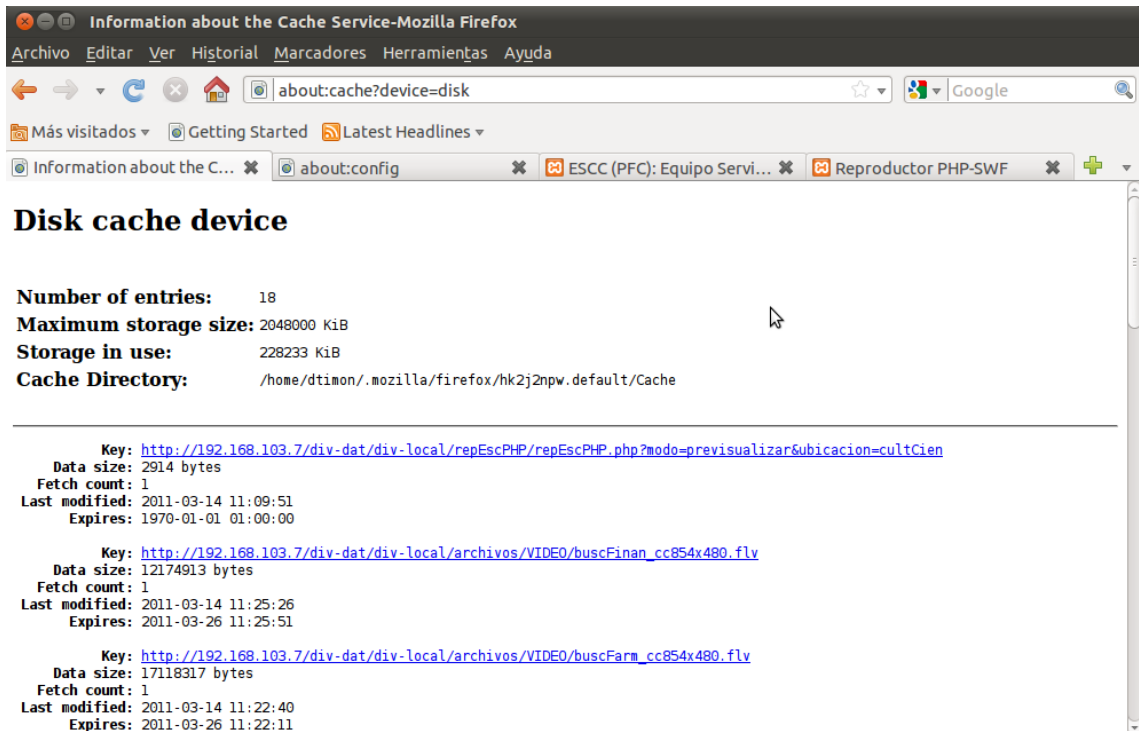


Ilustración 87.-. Archivos almacenados en disco. Arquitectura *caché* Auto-Proxy implementada con la configuración del navegador firefox.

El navegador, cada vez que tiene que reproducir un archivo, (cada vez que se visita la página o se realiza una petición desde las *aplicaciones swf*), comprueba si lo tiene almacenado en la *caché*. Si es así, el sistema lo descarga directamente del disco del propio equipo y si no vuelve a realizar una petición de este archivo al servidor desde donde lo descargó. Por tanto, se observa que con una correcta configuración del navegador podría implementarse un sistema similar al denominado *Auto-Proxy*.

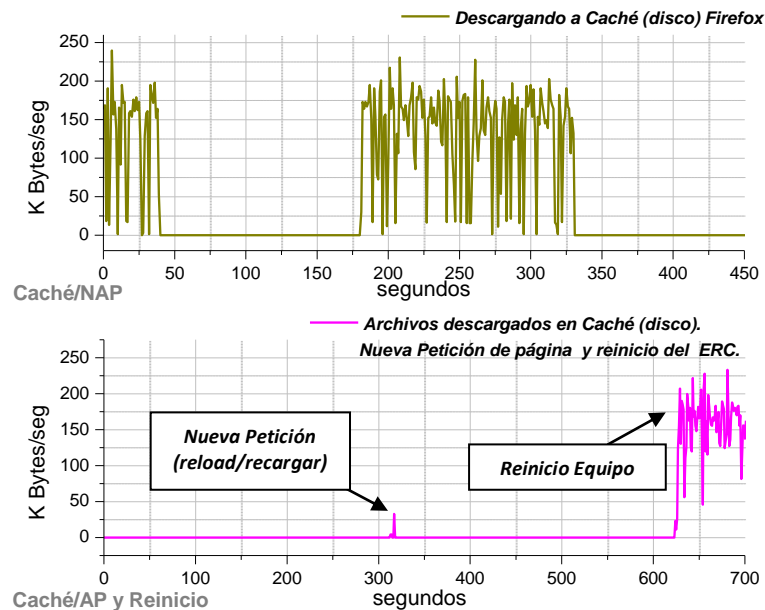


Ilustración 88.-. Captura obtenida para pruebas T002.

En la parte superior de Ilustración 88, se presenta la captura de red obtenida cuando el ERC no está implementando el sistema *Auto-Proxy* y se ha configurado la *caché* del navegador y, sin embargo, aún no ha almacenado de forma local los archivos porque es la primera vez que ha iniciado la reproducción. El comportamiento del equipo cuando está descargando los archivos es similar al de cualquiera de las pruebas en las que se ha forzado a que no se aplique la *arquitectura Auto-Proxy*; es decir, se establece una descarga progresiva mediante *HTTP Streaming*. Una vez la *lista de archivos (Lista_3_high: genUc3mPruebHQ, ver Tabla 20)* se ha reproducido por completo, se observa que el sistema, no vuelve a descargar los archivos para la reproducción, incluso cuando se realizan nuevas peticiones de la página (ver gráfica inferior de la Ilustración 88), de modo que se verifica el comportamiento de la *caché*.

No obstante, una vez se ha comprobado el funcionamiento aparentemente correcto mediante características inherentes al navegador utilizado, se ha observado que al sobrescribir el archivo en el *ESCC*, el equipo que está utilizando la *caché* del navegador no detecta estos cambios con la configuración suministrada. Es decir, en este caso el navegador no es capaz de diferenciar que el archivo ha sido modificado en el servidor. Además, cada vez que se inicia el equipo en el navegador se produce de nuevo la descarga de estos archivos según la configuración aplicada. Esta es una de las razones por la que se desarrolla una estrategia diferente para implementar la arquitectura denominada *Auto-Proxy* (ver apdo. 4.3.2 Arquitectura Funcional Auto-Proxy y 5.4.4.4 Desarrollo Arquitectura Auto-Proxy). También, y como argumento más importante para implementar la *arquitectura Auto-Proxy*, tal y como se ha definido, si el equipo cambiara de *Ubicación* o se cambiará la *lista de reproducción* de la *Ubicación* en un determinado instante, los *ERCs* debería descargarse los nuevos archivos de forma simultánea. Esto podría provocar un tráfico muy elevado en ciertas situaciones que podría sobrecargar la red y el mal funcionamiento en cada uno de los *ERCs* que están asociados a la *Ubicación* afectada.

9.3.3 Prueba Transversal Subjetiva de Calidad de Imagen en función de la resolución y del ancho de banda necesario en una sede. T003.

Independientemente del análisis del tráfico capturado, se puede realizar una valoración en términos subjetivos en lo relativo a la calidad percibida en la reproducción de los archivos. Para ello, se evalúa una de las *ubicaciones* en varios *escenarios* configurados, los cuales dependerán de la calidad de los vídeos de las *listas de reproducción*; es decir, los archivos mostrados serán los mismos, pero almacenados con distintas calidades. Así, para establecer una comparación subjetiva, aunque coherente, se han elegido vídeos para los cuales se dispusiera el mismo contenido en las tres calidades evaluadas en el proyecto fin de carrera: *LQ (Low Quality: Baja Calidad)*, *MQ (Medium Quality: Calidad Media)* y *HQ (High Quality: Calidad Alta)*.

Se ha elegido las *Lista_3* (ver Tabla 19 y Tabla 20), configurada en la *Ubicación* denominada *genUC3M*, la cual presenta las siguientes características de los archivos en cada modo configurado:

Archivo	Descripción	genUc3mPruebLQ	genUc3mPruebMQ	genUc3mPruebHQ
1	Entrevista a Hilary Putnam en la Universidad Carlos III de Madrid.	400x226/16:9/H.263 29,970fps/constante (256Kbps)	640x360/16:9/AVC(Main@L3.0) 29,970fps (YUV)/constante (417Kbps)	854x480/16:9/AVC(Main@L3.1) 30fps (YUV)/constante (910Kbps)
2	Espacio de Estudiantes de la Universidad Carlos III de Madrid.	400x226/16:9/H.263 29,970fps/constante (246Kbps)	640x360/16:9/AVC(Main@L3.0) 29,970fps (YUV)/constante (460Kbps)	854x480/16:9/AVC(Main@L3.1) 29,970fps (YUV)/constante (936Kbps)

Tabla 33.-- Resumen de archivos evaluados en las pruebas subjetivas.

En las pruebas que se muestran en este apartado se han adquirido con una resolución de pantalla de 1280x800px. Para ello, se ha hecho uso del mecanismo de previsualización de la *Ubicación* a pantalla completa y se ha pulsado la tecla *PrtSc* (*Print Screen*, *Imprimir Pantalla*) para copiar la imagen del vídeo emitido.

Los resultados de este procedimiento se muestran en los siguientes sub apartados:

- 9.3.3.1 T003LQ: Calidad Baja de Imagen en los Archivos Reproducidos
- 9.3.3.2 T003MQ: Calidad Media de Imagen en los Archivos Reproducidos
- 9.3.3.3 T003HQ: Calidad Alta de Imagen en los Archivos Reproducidos

Para seleccionar la configuración del escenario se ha partido de la resolución del *SPC* (*Sistema de Presentación de Contenido*) en la que se han presentado los vídeos, la cual era de 1280x800px. La relación (*ratio*) de los vídeos utilizados se muestra en la siguiente tabla:

Resolución	Coeficiente ratio	Ratio
LQ		
400x226	1,7699	≈16:9
MQ		
640x360	1,7777	≈16:9
HQ		
854x480	1,779	≈16:9
SPC		
1280x800	1,6	16:10

Tabla 34.-. Coeficientes y ratios según la resolución de los vídeos y la pantalla.

Teniendo en cuenta estos datos se observa que la resolución de la pantalla (*SPC*) mantiene el ratio 16:10, mientras que los vídeos es de 16:9. Así, se puede plantear sencillas ecuaciones, fijando algunos parámetros de inicio según el objetivo, el cual es resolver el porcentaje (tamaño) de la zonas que se debe establecer en la configuración del *escenario*: $Ancho(P_x)$ y $Alto(P_y)$.

Teniendo en cuenta las variables presentadas en la siguiente ilustración:

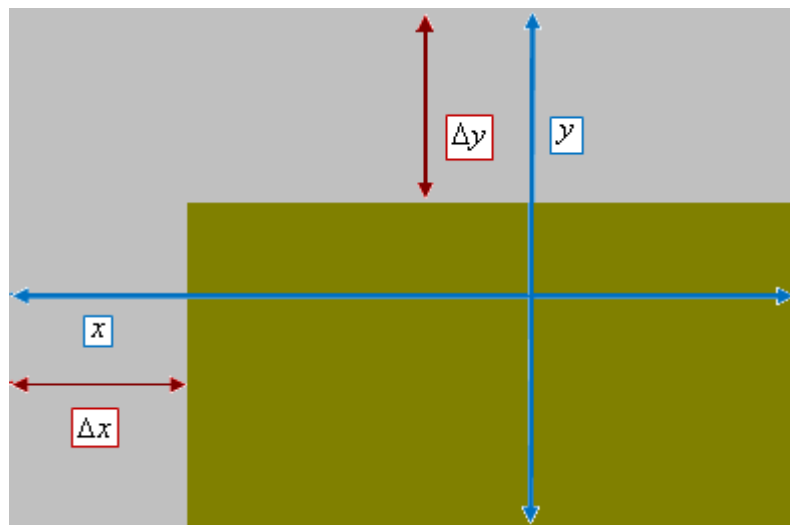


Ilustración 89.-. Resolución de SPC y Zona configurada para presentar los vídeos en las pruebas.

Y siendo en el SPC la resolución:

$$x = 1280$$

$$y = 800$$

Se puede establecer el objetivo inicial de que el margen de porcentaje horizontal (P_x) se mantenga con respecto al ancho total del SPC. En las pruebas se mantiene que:

- La resolución de la *Zona* sea menor o igual que la del vídeo de mayor calidad (HQ) y que sea mayor para el resto de calidades evaluadas (MQ y LQ).
- Establecer el vídeo con el mayor tamaño posible y la menor deformación geométrica³⁸ posible.

Por tanto, según los objetivo marcado debe cumplir, al menos, las siguientes restricciones:

$$360px(MQ) < y - \Delta y \leq 480px(HQ) \quad 640px(MQ) < x - \Delta x \leq 854px(HQ)$$

$$\frac{x - \Delta x}{y - \Delta y} \approx 1,77 \quad (\text{ratio } 16:9)$$

De la Ilustración 89 se deducen las siguientes ecuaciones que deben cumplirse:

$$\rho_y = \frac{y - \Delta y}{y} \quad (P_y = 100\rho_y) \quad \rho_x = \frac{x - \Delta x}{x} \quad (P_x = 100\rho_x)$$

De este modo se puede calcular el *porcentaje horizontal* con respecto al tamaño total de la pantalla (*Ancho*), de acuerdo al vídeo de mayor calidad; es decir, 854 px de resolución. Se tiene, por tanto, el límite del porcentaje para configurar la *Zona* ancha de la pantalla (en % del total) como:

$$\rho_x = \frac{854}{1280} = 0,6671 \quad (P_x = 66,71\%)$$

Como la configuración del sistema se ha establecido en saltos de 5% y se debe cumplir $x - \Delta x < 854px(HQ)$, la fijamos al 65 %, por lo que se tiene:

$$x - \Delta x = 0,65 \cdot 1280 = 832px < 854px$$

Además se debe de cumplir, debido a la resolución del vídeo, que:

$$\frac{x - \Delta x}{y - \Delta y} = 1,77 \quad (\text{ratio } 16:9)$$

Se deduce para el eje vertical, es decir, el *alto* de la *Zona*, el siguiente porcentaje:

$$\rho_y = \frac{x - \Delta x}{y \cdot 1,77} = 0,5875 \quad (P_y \approx 58,75\%)$$

³⁸ Existirá una deformación geométrica debido a que se ha marcado funcionalmente que el tamaño de las zonas (*Ancho*, *Alto*) sea configurado a intervalos del 5%.

Como se ha marcado que la configuración se establece en saltos del 5% (aunque modificando la base de datos directamente podría establecerse el valor deseado), se verifican los valores que cumplen la restricción en su redondeo y se observa:

P_y	$y - \Delta y$	Ratio
58,75%	470	1,77 ($\approx 16:9$)
55%	440	1,89
60%	480	1,73

Tabla 35.-. Elección porcentaje vertical.

Teniendo en cuenta el objetivo de presentar los vídeos de acuerdo a un *Marco* que establezca una resolución apropiada para pruebas subjetivas, se tiene que los porcentajes escogidos para desvirtuar geoméricamente lo menos posible los vídeos son, por tanto, *Ancho*=65% y *Alto* 60%. De este modo se tiene que el vídeo de *HQ* no aparecerá desvirtuado por aspectos relacionados con resolución y los vídeos de *MQ* y *LQ* teóricamente deberían sufrir interpolación, debido a que la resolución de la *Zona* donde se presentan es mayor que la resolución del propio vídeo.

A continuación, se presenta la configuración de todos los *escenarios* (se han configurado de igual modo: *Esc-0*, *Esc-1* y *Esc-2*) de la *ubicación* (*genUC3M*) para las pruebas establecidas con las distintas resoluciones en los archivos difundidos:

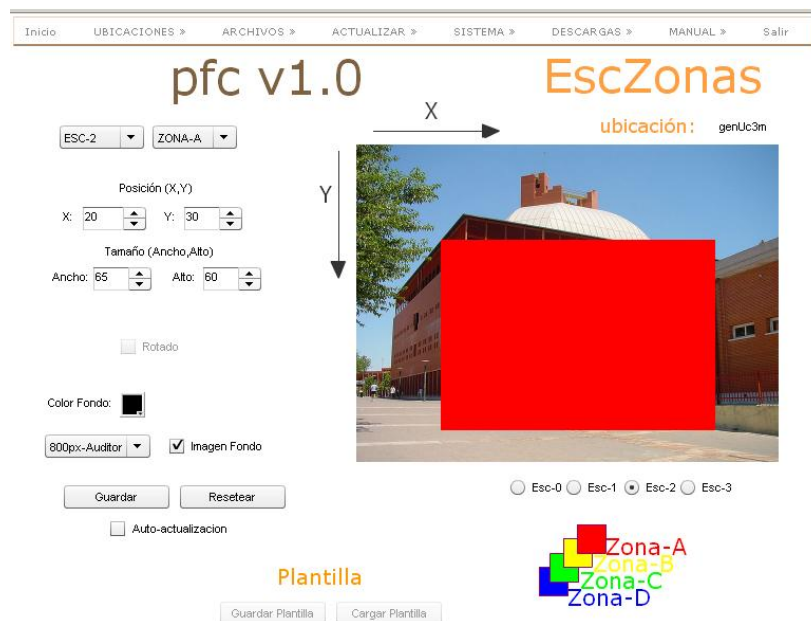


Ilustración 90.-. Configuración de la Ubicación. Pantalla Zonas a Escenarios.

Como se ha comentado, la resolución del SPC (*Sistema de Presentación de Contenido*) es de 1280×800 px. De este modo, la configuración de los escenarios se muestra al 65% y 60% de la resolución horizontal y vertical respectivamente (*Ancho*, *Alto*), por lo que se tiene un encuadre (*Zona*) para establecer los vídeos de 832×480 px de resolución. Esta resolución es suficiente, por tanto, para poder apreciar mejoras subjetivas de calidad, ya que la máxima resolución del vídeo en *HQ* es de 854×480 px, mientras que las resoluciones en la que se espera que las imágenes mantengan resolución defectuosa son de 640×360 px para la lista con archivos *MQ* y de 400×226 px para *LQ*.

El resultado obtenido en términos subjetivos ha sido el siguiente:

Tipo de Calidad	Observaciones
<i>LQ</i>	Cuando se incluyen vídeos de una calidad menor al soporte en el que son presentados, se aprecia una distorsión en la visualización de estos vídeos.
<i>MQ</i>	Al aumentar la calidad se observa que desaparecen gran cantidad de los defectos que se aprecian al emitir archivos de <i>LQ</i> . Sin embargo, al acercarnos físicamente al <i>SPC</i> para la visualización del vídeo en detalle, se pueden apreciar algunos defectos en los bordes observándose a veces pixelado.
<i>HQ</i>	Cuando la calidad del vídeo supera al del soporte en el que se difunde el contenido multimedia el vídeo aparentemente no muestra degradación en la presentación verificándose un perfecto visualizado.

Tabla 36.-. Resultados subjetivos obtenidos en visualización de vídeos con diferentes calidades (resoluciones).

Debido a estas observaciones es posible que en ciertas situaciones sea conveniente disminuir la calidad de los vídeos (y por tanto el ancho de banda utilizado) en según cómo se establezca la presentación de los vídeos hacia el público objetivo. A determinadas distancias el ojo humano no puede discriminar diferencias próximas en la calidad de los vídeos. Otra alternativa podría ser establecer la resolución del *SPC* acorde a la calidad de los vídeos difundidos para evitar que aparezca *pixelado* sobre los vídeos.

En los siguientes subapartados se muestran capturas³⁹ en las situaciones descritas en el apartado.

9.3.3.1 T003LQ: Calidad Baja de Imagen en los Archivos Reproducidos

Muestra de la Entrevista a Hilary Putnam en la Universidad Carlos III de Madrid



³⁹ Es posible que la impresión de las imágenes no permite verificar estas diferencias (sobre todo entre *MQ* y *HQ*); sin embargo, es posible apreciarlas sobre el archivo del documento en formato digital.



Muestra del vídeo Espacio Estudiantes de la Universidad Carlos III de Madrid.



9.3.3.2 T003MQ: Calidad Media de Imagen en los Archivos Reproducidos

Muestra de la Entrevista a Hilary Putnam en la Universidad Carlos III de Madrid





Muestra del vídeo Espacio Estudiantes de la Universidad Carlos III de Madrid.





9.3.3.3 T003HQ: Calidad Alta de Imagen en los Archivos Reproducidos

Muestra de la Entrevista a Hilary Putnam en la Universidad Carlos III de Madrid





Muestra del vídeo Espacio Estudiantes de la Universidad Carlos III de Madrid





9.4 Resumen

En este capítulo hemos presentado los resultados de las pruebas realizadas sobre el *entorno de pruebas* presentado en el Capítulo 8.

Sobre este entorno hemos realizado un buen número de pruebas para estudiar el comportamiento de la aplicación de *Cartelería Digital* sobre el *Marco* de forma rigurosa. También se han realizado otras pruebas que evidencian la utilidad del *Marco* para otras aplicaciones de *Difusión Selectiva* que requieran reproducción de contenido *Streaming Media* cuyas modificaciones necesarias se han presentado en el Capítulo 6.

En las pruebas se ha verificado el funcionamiento de la *arquitectura funcional* y, debido a la realización de las pruebas, también se ha llegado a una serie de conclusiones:

- El funcionamiento del *Marco* puede verse afectado por la emulación del *entorno de pruebas* con un solo *equipo anfitrión* donde se ejecuten varios equipos virtualizados y herramientas de análisis de forma simultánea. Esta degradación del sistema se debe al uso compartido de los recursos: *CPU, memoria, disco duro,...*
- Se ha mostrado cómo el *entorno de pruebas* se puede construir mediante una combinación moderada de uso de *hardware* físico y aplicaciones para virtualización.
- La configuración del *entorno de pruebas* en lo relativo a la simulación de la *WAN* ha sido apropiada para emular diferentes sedes conectadas a una sede principal con un número de equipos de red limitado e inferior a los que inicialmente cabría esperar.
- Se ha podido comprobar que el desarrollo de una *arquitectura funcional Auto-Proxy* sobre el *Marco* es una excelente elección, ya que no se genera apenas tráfico entre los equipos en una situación estable de funcionamiento.
- El *Marco* desarrollado, en la arquitectura *Auto-Proxy*, puede establecer en condiciones inadecuadas de configuración la peor situación de funcionamiento (*NAP/CS, No Auto-Proxy/Con Sincronización*) de forma transitoria: presencia, de forma simultánea, de tráfico *HTTP Streaming* y de tráfico debido a la sincronización de contenido multimedia.
- La restricción de la utilización de un *ancho de banda* para la sincronización del contenido ha sido adecuada para evitar la saturación de la red cuando se carga nuevo contenido multimedia en el servidor.
- Se ha concluido que, en una arquitectura *cliente-servidor* para una aplicación de *Difusión Selectiva*, es necesario el desarrollo de algún mecanismo para salvaguardar el *ancho de banda*, ya que se han observado una serie de desventajas en la *arquitectura funcional* en ausencia de la característica *Auto-Proxy*:
 - Configurar varias *zonas* sobre un *Escenario* de una *Ubicación* es equivalente a introducir nuevos equipos *ERC* de una sola *Zona* sobre la misma sede. Por tanto, las necesidades de una sede es la suma del *ancho de banda* necesario para reproducir el archivo de mayor necesidad de todas las *zonas* de todos los *ERCs* de la sede.
 - Cuando las necesidades de *ancho de banda* de una sede son mayores que el *ancho de banda WAN* disponible en la sede, el funcionamiento de la aplicación de *Cartelería Digital* falla en la sede y puede ralentizar la reproducción del contenido multimedia en todos los *ERCs* de la sede.
 - Cuando las necesidades de *ancho de banda* de la suma de todas las sedes es superior al *ancho de banda WAN* disponible desde la sede principal, el

funcionamiento de la aplicación de *Cartelería Digital* falla y puede producirse un funcionamiento anómalo en todos los *ERCs* del sistema.

- Se ha comprobado que añadir nuevas funcionalidades y aplicaciones al *Marco* desarrollado mediante el uso de *plugins* sobre el navegador *Mozilla Firefox* es una estrategia adecuada y sencilla. Esto muestra la evidencia de la flexibilidad y versatilidad del *Marco* desarrollado por lo que pueden añadirse nuevas de *aplicaciones de Difusión Selectiva*.
- Las características de las funcionalidades incluidas con la utilización de *plugins* en el *Marco* desarrollado depende de las funcionalidades del propio *plugin*. Se ha verificado la idoneidad el uso del *plugin VLC* para *Mozilla* permitiendo la reproducción de flujos *Streaming Media Multicast* usando *RTP/MPEG-TS* para el transporte del contenido multimedia incluso en directo.
- La emisión en modo *Streaming Media Multicast* mediante la *aplicación VLC* permite ajustar el códec de audio y video en la emisión, por lo que no es posible obtener información acerca de las características del archivo emitido.
- Hemos verificado la interacción entre los equipos principales (*ESCC* y *ERC*) de acuerdo a la *arquitectura funcional* diseñada en el Capítulo 4, verificando que la secuencia de inicio realiza el intercambio de paquetes esperado.
- Hemos configurado la memoria *caché* del navegador *Mozilla Firefox*, con la intención de obtener un funcionamiento similar a la arquitectura funcional *Auto-Proxy*. En la configuración se ha detectado parte de la viabilidad de este mecanismo y también se ha comprobado la ausencia de tráfico relacionado con la sincronización de contenido. Sin embargo, debido a la naturaleza de la aplicación de *Cartelería Digital* y al alto tráfico que podría generarse en cambios en la configuración de los *ERCs* (cambios de *listas de reproducción* programadas) se ha evidenciado la necesidad del mecanismo *Auto-Proxy* desarrollado en el *Marco*.
- Hemos visto la necesidad de tener en consideración la resolución de los vídeos en la configuración de las *zonas* del *Escenario* y la relación que existe con la resolución del *SPC*. Estas consideraciones se debe tener en cuenta para configurar el *Escenario* y generar los videos de acuerdo a unas dimensiones de resolución.

PARTE V. Conclusiones y Trabajos Futuros

Capítulo 10 Conclusiones y Trabajos Futuros

En este capítulo se presentan las conclusiones obtenidas al finalizar el desarrollo del *Marco* diseñado para *Aplicaciones de Difusión Selectiva de Contenido Multimedia* y se muestran algunas *líneas de trabajo futuro* que se podrían llevar a cabo como continuidad del trabajo realizado.

10.1 Conclusiones

En este trabajo se ha realizado un estudio de los diversos mecanismos existentes en el mercado para proveer *Difusión Selectiva (Narrowcasting)*. Este tipo de soluciones se posicionan como un elemento eficiente para la difusión de información en entornos como en empresas y el mundo de la publicidad, con ventajas de flexibilidad, dinamismo, y atractivo frente a soluciones tradicionales. Pero para conseguir estas ventajas son necesarias las herramientas que permitan crear estas aplicaciones de forma sencilla. Este proyecto ha contribuido a este objetivo diseñando y desarrollando un *Marco* para la creación de aplicaciones de difusión selectiva con un aprovechamiento de una *arquitectura funcional* definida y bajo elementos desarrollados e integrados que permiten introducir nuevas funcionalidades realizando pequeñas modificaciones.

En la definición considerada de *Difusión Selectiva*, los criterios o estándares de cómo obtener el servicio los establece cada fabricante según su criterio. Como parte del trabajo se han estudiado de forma superficial, pero suficiente, un buen número de soluciones de *Difusión Selectiva* disponibles en el mercado. Se ha encontrado que la aplicación con mayor peso en el mercado es la de proporcionar *Cartelería Digital*; sin embargo, no es la única aplicación presente en las soluciones comerciales. Se optó por considerar la solución de *Difusión Selectiva* ofrecida por Cisco (*DMS, Digital Media System*) [134] como una solución de referencia para nuestro trabajo, ya que ofrece otras aplicaciones además de la de *Cartelería Digital*.

Entre las características más destacables de la solución de referencia se tienen:

- Es una solución de propósito general y ofrece varias aplicaciones de *Difusión Selectiva*: *Cartelería Digital (Digital Signage)*, *Portal de Video (Video Portal)* y *Sistema de TV Corporativo (Enterprise TV)*.
- Proporciona gestión centralizada mediante varias interfaces web que convierten la configuración de todo el sistema en un procedimiento sencillo.
- Soporta acceso a archivos de audio y vídeo bajo demanda (*VoD*).
- Soporta *difusión de contenido en directo (Live Streaming)*, tanto *Unicast*, como *Multicast*.
- Reduce el impacto del *ancho de banda* de vídeo en la red, mediante estrategias (*ACNS, WAE*) y protocolos (*WCCP*) que permiten utilizar el *ancho de banda* de forma eficiente para todas las aplicaciones que soporta.
- Ofrece *Difusión Selectiva Interactiva* en la aplicación de *Sistema de TV Corporativo* mediante la selección de canales con un mando a distancia.
- Permite reproducción de contenido multimedia alternativo ante caídas del enlace.

Adicionalmente a estas características se han observado algunas desventajas en la solución de Cisco (DMS):

- Está compuesta por un número elevado de equipos que forman la solución con respecto a otras soluciones.
- El número y tipo de elementos varía ante la ausencia o presencia de una WAN, según la solución de referencia opere en un contexto de *Espacio de Funcionamiento Simple o Distribuido*.
- En un *Espacio de Funcionamiento Distribuido*, es necesario que los elementos de red principales de la sede también sean del mismo fabricante de referencia [135], lo que la convierte en una solución bastante cerrada y restringida al uso de una tecnología.

En el presente trabajo se ha desarrollado un *Marco para Aplicaciones de Difusión Selectiva* que proporciona las características más interesantes que se han observado en la mayoría de soluciones estudiadas, supliendo las deficiencias encontradas; por tanto, la solución de referencia ha sido útil para identificar una serie de características que debía cumplir el sistema desarrollado.

Hemos concluido que el diseño del *Marco para Aplicaciones de Difusión Selectiva* viene supeditado a la definición inicial de una *arquitectura funcional* que, de forma explícita, indique el funcionamiento lógico de cada elemento dentro del sistema a implementar. A esta *arquitectura funcional* definida la hemos denominado *arquitectura Auto-Proxy*.

De este modo, el *Marco* se ha diseñado con una doble función: permitir la configuración y el contenido multimedia a los equipos y establecer la reproducción del contenido de acuerdo a la configuración y la *arquitectura funcional*. Las características principales del *Marco* sobre la *arquitectura funcional Auto-Proxy* definida son:

- Arquitectura *cliente-servidor* en el que la configuración y el contenido multimedia se gestionen de forma centralizada en un único equipo. Hemos considerado que este mecanismo es el más idóneo debido a que clarifica y facilita el diseño del sistema.
- Preservación el *ancho de banda* disponible. La congestión del tráfico ha sido siempre un problema evidente en el paradigma *cliente-servidor*; sin embargo, con las medidas tomadas (*arquitectura Auto-Proxy* y limitación de *ancho de banda* en la sincronización del contenido) se resuelven estos problemas de forma eficiente.
- Disponibilidad ante fallos de la red o equipo servidor (ESCC). Una arquitectura *cliente-servidor* muestra un claro punto de fallo, el cual es el propio equipo servidor (ESCC); no obstante, el hecho de mantener el contenido multimedia almacenado en los *equipos terminales* permite el funcionamiento del sistema ante cortes en la disponibilidad al equipo principal (ESCC) ya sea por fallos en la red o del propio servidor, por lo que nuevamente se supera esta dificultad de las arquitecturas *cliente-servidor*.
- Diseño y desarrollo del *Marco* en diferentes *Capas* que permiten flexibilidad e independencia tecnológica entre *Capas*. Además, el hecho de diferenciar el sistema en *capas* permite que, en el futuro, cada *Capa* se pueda ejecutar en diferentes equipos permitiendo una alta escalabilidad. Otras ventajas significativas de la utilización de *capas* para el desarrollo son:
 - Desarrollo en Paralelo. Cada *Capa* puede ser desarrollada en paralelo de forma independiente unas de otras, permitiendo que los equipos de desarrollo se puedan dividir en tareas.

- Facilidad al Cambio. La implementación de cada *Capa* puede ser sustituida por otra tecnología que cumpla las funcionalidades necesarias y que pudieran surgir o aportar otras ventajas: mayor rapidez, eficiencia,...
 - Aplicaciones fáciles de Testear. Cada *Capa* se puede probar de forma independiente lo cual facilita la resolución de problemas.
 - Reutilización. Los módulos de cada *Capa* se pueden reutilizar en otras aplicaciones.
 - Facilidad de Comprensión. Si una aplicación está formada por varias *capas*, resulta más sencilla la comprensión y el estudio de cada *Capa* de forma individual para comprender la aplicación en su conjunto.
- Utilización de tecnologías web tanto para configurar el sistema como para la reproducción de contenido, lo que simplifica el desarrollo y despliegue del sistema. También, el hecho de haber desarrollado una aplicación web para gestionar la configuración del sistema permite que el usuario administrador pueda acceder a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como pulsar botones y rellenar y enviar formularios. La utilización de una *aplicación web*, para la configuración como para la reproducción, aporta otra serie de ventajas menos evidentes:
 - Se pueden realizar tareas sencillas sin descargar ni instalar ningún programa y son portables ya que se ejecutan en un navegador web.
 - Carece de problemas de compatibilidad en función del *sistema operativo* (*Windows, Linux, OS X, etc.*) desde donde se realizan las configuraciones (*EACCC*).
 - Los posibles virus del equipo que realiza la configuración (*EACCC*) no dañan el sistema ya que la aplicación se encuentra de forma centralizada en el servidor (*ESCC*).
 - La *aplicación web* no ocupa espacio en el disco duro del equipo desde el que se realiza la configuración (*EACCC*).
 - El consumo de recursos en la configuración es moderado ya que parte de la aplicación se ejecuta directamente en el servidor (*ESCC*).
 - La actualización de la aplicación, tanto para la configuración como para la reproducción, es inmediata ya que sólo se ha de incluir los nuevos desarrollos en el servidor (*ESCC*).
- Propósito general para soportar múltiples aplicaciones además de la aplicación de *Cartelería Digital*. Resulta una característica fundamental, ya que la solución permite otro tipo de aplicaciones que pueden estar basadas en otros parámetros diferenciadores de *Difusión Selectiva* y permite adaptar el *Marco* a las aplicaciones que se requieran según la tendencia del mercado de forma sencilla.
- Comunicación entre el servidor y los equipos terminales ante cambios de configuración. Esta característica permite una comunicación eficaz y se realiza mediante un mecanismo basado en *socket*. Las ventajas que encontramos al utilizar *sockets* es que se permite crear programas en esquemas *cliente-servidor* sin tener que preocuparse del tipo de comunicación que exista e independiza el lenguaje de programación utilizado en cada una de las máquinas remotas. Mediante *socket* se puede intercambiar cualquier flujo de datos de manera fiable y ordenada, de modo que es idóneo para transmitir cambios en la configuración.
- Independencia de la arquitectura de red y del tipo de conexión de los elementos. Esta característica independiza la configuración de las aplicaciones de *Difusión Selectiva* de la configuración de la red, lo que permite simplificar ambos procesos de configuración.

Para alcanzar las características del *Marco* mencionadas se ha realizado un estudio de las diferentes alternativas tecnológicas para desarrollarlo y se ha tenido que realizar un estudio en mayor profundidad para manejar las tecnologías seleccionadas en cada una de las *capas*. Las principales tecnologías web utilizadas para el desarrollo del *Marco* mediante *capas* ha sido *Flash* (*ActionScript 3*) junto con *PHP* y *MySQL*, ejecutadas en sistemas operativos basados en *Linux*, por lo que se ha hecho necesario instalar en los equipos un *servidor Apache*, *MySQL* y *PHP*, además de algunos *plugins* y complementos sobre los navegadores para proporcionar reproducción del contenido:

- La elección finalmente de una distribución del sistema *Linux* (*Ubuntu Server 10.10* y *Ubuntu Desktop 10.10*)[136] sobre los equipos principales ha permitido establecer un sistema robusto con algunos servicios ausentes en otras plataformas de forma gratuita.
- Las aplicaciones *Apache*, *MySQL* y *PHP* se ha instalado mediante la aplicación empaquetadora *XAMPP* [91], la cual facilita mucho la instalación y administración de las aplicaciones independientemente del sistema operativo utilizado.
- El uso de *php* ha sido una excelente elección debido a que los *script* programados en este lenguaje pueden ser utilizados, tanto para generar páginas dinámicas, servir las *aplicaciones swf*, implementar mecanismos *asíncronos* (*Motor PHP*), escribir aplicaciones (*servidores socket*) o ser utilizados para realizar llamadas a otras aplicaciones (*script de inicio*).
- La elección de una tecnología como *ActionScript 3* y *Adobe Flash Player* para desarrollar y ejecutar *aplicaciones swf*, ha sido una elección adecuada debido a las facilidades que establecen para introducir contenido multimedia, además de proveer un mecanismo para establecer la comunicación con el servidor a través de *sockets*[137].

Debido al diseño del *Marco*, las tecnologías utilizadas para llevarlo a cabo y la aplicación de *Cartelería Digital* desarrollada para testear el *Marco* se han observado algunas particularidades que condicionan el funcionamiento del sistema. Estas particularidades se deben tener presentes tanto para la modificación y/mejoras del *Marco*, como para la introducción de nuevas aplicaciones sobre éste:

Streaming y Reproducción de Contenido Multimedia

Se ha observado que el término *Streaming* es un concepto muy utilizado en *Difusión Selectiva*; sin embargo, el concepto queda diluido en varios mecanismos de transporte de contenido multimedia, tal y como se ha diferenciado en el presente trabajo (Capítulo 2): *Streaming Media* y *Pseudo Streaming*.

La aplicación de *Cartelería Digital* que se ha desarrollado soporta *HTTP Streaming* (*Pseudo Streaming*) y se ha considerado que es suficiente ya que permite la difusión de contenido multimedia de forma adecuada y el uso del protocolo *http* facilita la configuración, evitando incluir nuevos servidores en la *arquitectura funcional* como se ha considerado en el Capítulo 5.

La implementación de la aplicación mediante métodos *HTTP Streaming* y *aplicaciones swf* acota la reproducción de contenido a ciertos contenedores multimedia (*flv*), limitando la codificación de los archivos a determinados *códec* (ver apdo. 2.6.3); no obstante, el *Marco* permite cambiar el reproductor del contenido multimedia a otras tecnologías que permitan otros contenedores multimedia sin variar el resto del *Marco*. Desarrollar un sistema basado en *HTTP Streaming* limita la emisión de contenido en directo; sin embargo, esta limitación carece de repercusión en una aplicación de *Cartelería Digital*. Además, con las mejoras introducidas

(Capítulo 6) hemos mostrado que extender el *Marco*, para añadir otros tipos de aplicaciones y de *Streaming*, es fácil y de hecho se ha demostrado la viabilidad de hacerlo (ver después, *Mejoras del Marco*).

Arquitectura Funcional

Incluir una *WAN* entre el equipo servidor y los equipos reproductores no requiere incluir nuevos elementos en la arquitectura funcional implementada; sin embargo, sí lo es para un correcto funcionamiento de la *arquitectura de referencia*. Esta característica hace adecuada la solución planteada y desarrollada, para esquemas en los que no se quiera modificar la red de la organización y es una ventaja considerable para *espacios de trabajo* en el que se tengan un número significativo de sedes.

En el diseño del *Marco* se estableció la necesidad de mantener la hora sincronizada en los equipos y, aunque se ha obtenido mediante aplicaciones y protocolos ya desarrollados y ampliamente testados (*NTP*) [102] [138], sin la sincronización de hora, se produciría un funcionamiento inadecuado en la aplicación de *Cartelería Digital* sobre el *Marco* diseñado. Es por ello que se ha de considerar la sincronización de hora como una necesidad funcional de la arquitectura. También, se ha de considerar configurar algunos elementos de red adicionales relacionados con la seguridad, como son los posibles firewalls en la red, debido a que para el funcionamiento del sistema se hace uso de determinados puertos *UDP/TCP*.

Virtualización y Entorno de Pruebas

Establecer un entorno de pruebas mediante la combinación de equipos virtualizados con *VMware* [97] [55] [139] y *hardware* físico ha permitido simular espacios de trabajo lo suficientemente sofisticados para probar el *Marco* de forma bastante rigurosa con recursos limitados. Así, se ha concluido, también, que la virtualización es un mecanismo muy útil y flexible a la hora de simular arquitecturas complejas en laboratorios. En las pruebas realizadas hemos observado que la virtualización de equipos ha de ser moderada, ya que los sistemas se ven degradados al compartir los recursos de un *equipo anfitrión*. También se ha evidenciado, con la realización de diversas pruebas que nos permiten ver el comportamiento del sistema como si se careciera de la implementación de la arquitectura *Auto-Proxy*, la necesidad de incluir un mecanismo para salvaguardar el *ancho de banda* y nos ha permitido verificar que el *Marco* ideado posibilita que se ejecuten aplicaciones con una alta eficiencia en la utilización de este recurso escaso. Además, debido a que el *entorno de pruebas* se ha ajustado a una situación real de una típica organización con sedes distribuidas, hemos podido valorar los problemas que pueden surgir ante un dimensionamiento erróneo de la red, sobre todo en la sede principal, y ausencia de un mecanismo que limite el uso del *ancho de banda* en la utilización de *HTTP Streaming*.

Ancho de Banda y Arquitectura Auto-Proxy

Se ha experimentado en el funcionamiento del *Marco* sobre el entorno de pruebas la importancia de configurar un *ancho de banda* apropiado especialmente en la sede principal donde se encuentra el *ESCC*, hacia las sedes donde se encuentran los *ERCs*. En la aplicación de *Cartelería Digital*, un mal dimensionamiento de la red en lo relativo al *ancho de banda* provocaría que la reproducción de un vídeo llevara un transcurso temporal anormal en la descarga a través de *HTTP Streaming*; no obstante, al implementar la arquitectura *Auto-Proxy* es posible la reproducción de vídeos incluso si se carece del *ancho de banda* mínimo para reproducir el vídeo. La arquitectura *Auto-Proxy* en situación estable tiene la máxima eficiencia posible en la utilización de *ancho de banda*, como se ha observado en las pruebas realizadas

sobre una aplicación de *Cartelería Digital*. El uso eficiente del *ancho de banda* nos permite establecer un mayor número de sedes remotas (*ERCs*) que dependa del mismo servidor (*ESCC*) centralizado (arquitectura *cliente-servidor*), en la que se ha de distribuir un gran flujo de información como en el caso del sistema desarrollado. Con la *arquitectura funcional Auto-Proxy*, en una situación estable de funcionamiento, además de aumentar el número total de *ERCs* que dependen del *ESCC*, la configuración del número de *zonas* de reproducción de contenido de cada *ERC* tampoco influye en la utilización del *ancho de banda* como ocurre en la misma *arquitectura funcional* en ausencia de la característica *Auto-Proxy*.

Mejoras del Marco

En el diseño inicial de la *arquitectura funcional* y del *Marco* se ha utilizado *HTTP Streaming Unicast* descargado directamente desde el *ESCC* o el *ERC (Auto-Proxy)*, estableciendo inicialmente una aplicación de *Cartelería Digital*. Sin embargo, el *Marco* y la *arquitectura funcional* están preparados para asumir otras características adicionales como recuperar flujos *Streaming Media* y *Multicast*.

Se ha propuesto, en un capítulo independiente (Capítulo 6), un método para incluir estas funcionalidades (*Multicast*, *Streaming Media*, *Emisión en Directo*,...) de forma muy simple sin apenas modificar el *Marco*. Esta posibilidad es consecuencia directa del mecanismo de presentación del contenido y de las aplicaciones de terceros disponibles. Debido a que se ha desarrollado un *Marco* en el que se ha utilizado un *plugin (Flash Player)* que ejecuta *aplicaciones swf* desarrolladas para reproducir el contenido, se ha planteado incluir nuevos *plugins* que ejecuten aplicaciones de terceros. Una nueva funcionalidad puede ser la reproducción de contenido multimedia mediante *Streaming Media* con diversos protocolos como *RTP*, *RTSP* y utilizando *Multicast*, además de *Unicast*. Esto se ha conseguido utilizando el *plugin VLC* [111] de modo que el *Marco* puede sustentar otras aplicaciones como un *Sistema de TV Corporativo* incluso con contenido en directo.

Existen también otras aplicaciones de interés en *Difusión Selectiva*; un reproductor de *SMIL* [117] está disponible mediante el *plugin Ambulant* [118] para *Mozilla Firefox* en *Linux* e incluso se pueden desarrollar nuevos *plugins* sobre el navegador de acuerdo a otras posibles necesidades. Por tanto, el *Marco para Aplicaciones de Difusión Selectiva de Contenido Multimedia* desarrollado que implementa la *arquitectura funcional Auto-Proxy* es válido para aplicaciones de *Cartelería Digital* y, además, sienta las bases para añadir todas las aplicaciones provistas por la *arquitectura de referencia*, como son el *Portal Web de Vídeo* y el *Sistema de TV Corporativo* de forma aproximada a como lo define el fabricante *Cisco*.

Mediante el mismo mecanismo de la *arquitectura funcional Auto-Proxy* para el intercambio de información basado en *sockets*, se permite construir planteamientos más sofisticados de la *arquitectura funcional* como, por ejemplo, la integración de sensores mediante dispositivos electrónicos de bajo coste [140] [141] que puedan comunicarse mediante *sockets* [142] [143]. Por tanto, es reseñable también indicar, que mediante el *Marco para Aplicaciones de Difusión Selectiva* desarrollado, es posible, incluso, añadir interactividad de forma genérica y no sólo la interactividad de mando a distancia [128], obteniendo sistemas de *Difusión Selectiva Activa o Adaptativa* y *Sistemas de Difusión Selectiva Interactiva*. Esto se puede conseguir sin realizar excesivas modificaciones sobre el *Marco*, mediante la creación de un *socket local* en los *ERCs* que pueda comunicarse con aplicaciones que controlen los distintos sensores.

10.2 Líneas de Trabajo Futuro

Dentro del *Marco* desarrollado existen posibles *líneas de trabajo futuro* que podrían ir estableciendo una arquitectura funcional más completa y robusta, y un *Marco* más seguro que ofrezca mayores funcionalidades. Las opciones detalladas en este apartado son solo algunas de las muchas opciones que podrían seguirse y se enumeran para intentar hacer ver al lector las muchas posibilidades del *Marco* implementado. Las *líneas de trabajo futuro* que vamos a describir se centran sobre varios caminos diferentes: establecer nuevas funcionalidades a la aplicación de *Cartelería Digital* desarrollada, añadir nuevas aplicaciones al *Marco* (*Canal de TV Corporativa*, *Portal Web de Vídeo*), mejorar aspectos relacionados con la *arquitectura funcional*, añadir *Difusión Selectiva Interactiva*, y mejorar aspectos relacionados con la seguridad.

A continuación, se pasa a detallar estas posibles *líneas de trabajo futuro*, con ejemplos de tareas específicas para cada uno de los puntos mencionados anteriormente.

10.2.1 Nuevas Funcionalidades y Mejoras de la Aplicación de Cartelería Digital

Las nuevas funcionalidades añadirían mejoras y características adicionales a la *aplicación de Cartelería Digital*. El esfuerzo que debe realizarse en esta línea de trabajo estaría orientado a trabajar sobre la *Interfaz Gráfica de Usuario*, el *Motor PHP* y modificaciones en el *modelo de datos*. Entre ellas se podrían encaminar los esfuerzos para:

- *Introducir el concepto de Plantillas*. La funcionalidad permitiría guardar y cargar la configuración mediante plantillas. Se debería modificar el modelo de datos y las pantallas para permitir configuraciones basadas en plantillas. De este modo una vez configurados los *escenarios* de una *Ubicación*, podría copiarse la configuración establecida y almacenarse en un fichero para cargarlas en una nueva *Ubicación*.
- *Permitir programar el cambio del Escenario a partir de pantallas de programación*. La funcionalidad implementada en el *Marco* a través de la *Interfaz Gráfica de Usuario* ha sido que el usuario administrador puede configurar hasta 4 escenarios sobre cada *Ubicación*; sin embargo, sólo puede seleccionar uno de los escenarios de forma manual. Una mejora podría ser que estos escenarios se pudieran programar de forma automática conforme a unos criterios temporales. Algunas de las soluciones estudiadas permiten ir cambiando de configuración del *Escenario* a difundir, según un patrón horario, siendo días pares o impares, fines de semana,...
- *Permitir reproducir formatos adicionales de contenedores multimedia*. Actualmente, se permite subir al servidor diferentes tipos de archivos; sin embargo, el listado de los tipos de archivos permitidos es limitado (*.jpeg*, *.bmp*, *.gif*, *.flv*). Existen ciertas aplicaciones disponibles como *ffmpeg* [144] que permiten transformar los tipos de archivos a otros contenedores multimedia modificando, incluso, los *códec* con los que se ha codificado el vídeo y el audio. Debido a que en la implementación del sistema las *aplicaciones swf* sólo permiten la reproducción de archivos *.flv*, una estrategia muy interesante, para salvar esta limitación, podría ser realizar una llamada a la aplicación *ffmpeg* mediante el *script php* que recibe el archivo, al cargarlo en el servidor, con la intención de convertirlo al contenedor multimedia soportado (*flv*).
- *Actualización del ESCC a partir de la ejecución de un script*. Podría programarse la funcionalidad en el servidor que al cargar un script con los archivos de actualización se estableciera de forma automática las nuevas funcionalidades programadas. Para ello se debería programar un *script php* que recogiera el archivo empaquetado de actualización y que ejecutara los comandos necesarios (*cp*, *mv*, *rm*, *exec*,...). Con esta

funcionalidad el sistema se podría actualizar a nuevas versiones del software sin la necesidad de establecer de nuevo la configuración de todos los archivos descritos. En el proceso de actualización, se debería mantener la configuración de la versión anterior. Algunas aplicaciones de algunos fabricantes de *software* [145], como medida de seguridad, mantienen varias particiones en el proceso de actualización, de modo que cargan el nuevo *software* en una partición inactiva junto con la migración de la configuración y permite al administrador cambiar de una partición a otra.

- *Actualización los ERCs de forma automática.* Se podría implementar un mecanismo en los ERCs que detectaran la disponibilidad de una nueva versión del *software* para la configuración de los ERCs en el ESCC. Así, cuando el ERC se conecte al ESCC podría comprobar si tiene la última versión disponible en el ESCC y ejecutar la secuencia de actualización si fuera necesario.
- *Sistema de Acceso Multiusuario.* Muchos sistemas permiten definir usuarios y las páginas de acceso a las que se les permitirá acceder a dichos usuarios. Así, es interesante poder crear usuarios en el sistema que sólo permitan el acceso a la gestión de contenido, otros que permitan el acceso a la programación y otros a los parámetros de sistema. Esta funcionalidad no es difícil de obtener en el sistema desarrollado. En el desarrollo tan sólo se han creado dos usuarios. En función del usuario se mostrarían cada una de las opciones de configuración del sistema en función de privilegios asociados. Se debería proporcionar ciertas pantallas en la *Interfaz Gráfica de Usuario*, de modo que permitiera crear diferentes perfiles de acceso a la configuración, tanto en el ESCC, como en el ERC. El Marco podría integrarse incluso con un sistema LDAP que proporcionase la creación y/o autenticación de usuarios.
- *Configuración de hora a través de Interfaz Gráfica de Usuario.* En el sistema desarrollado, se ha establecido que la sincronización de hora se establezca a través de la sincronización de los ERCs, con el ESCC mediante el protocolo NTP y, a su vez, el ESCC se sincronice con una referencia NTP externa. Por tanto, se está presuponiendo que el servidor siempre tendrá acceso a una *referencia NTP externa*; sin embargo, en situaciones reales esta circunstancia no siempre es posible y, por tanto, resulta interesante permitir al usuario administrador del sistema, configurar la hora en el propio servidor ESCC de forma manual mediante una interfaz gráfica de usuario.
- *Interfaz para Configuración de Recepción de Contenido Streaming Media.* Tal y como se ha comentado en el Capítulo 6, es posible implementar un mecanismo en el que reproducir un flujo Multicast a partir de ciertos parámetros: *protocolo, dirección IPv4 y puerto (TCP/UDP)* en el que establecer la recepción del flujo. Así, como mejora es posible añadir a la *Interfaz Gráfica de Usuario* la configuración de estos parámetros en cada *Ubicación* para establecer un sistema de *Difusión Selectiva* que permita mostrar el *Streaming Media Multicast o Unicast* presente en la red.

También se podrían establecer mejoras en las funcionalidades atribuidas al *Reproductor de Zona y Control SWF*:

- *Comunicación entre el Control SWF y la Reproducción de Zona.* Debería poder permitir establecer un mecanismo de comunicación entre aplicaciones embebidas en el navegador para así actualizar sólo las zonas y no la recarga de toda la página. Otra alternativa sería establecer el control dentro de las aplicaciones de zonas pero entonces se deberían soportar varias conexiones al socket (una por cada zona)

Otras funcionalidades que podrían añadirse a la aplicación, estarían relacionadas con mejoras establecidas a nivel de configuración de sistema operativo:

- *Interprete de comandos o consola (Shell) específico.* El desarrollo de una aplicación que sustituyera al *Shell* por defecto (*bash*), resultaría muy útil como prevención ante posibles fallos de las aplicaciones *Apache*, *php* o *MySQL*, permitiendo el acceso a unas pocas aplicaciones. De esta forma, mediante la aplicación futura desarrollada, se podrían realizar algunas acciones en situaciones críticas que permitan volver el sistema de nuevo estable: reiniciar equipo, reiniciar ciertos servicios, cambios de configuraciones de interfaces de red,... En el desarrollo del proyecto, se ha utilizado la siguiente *Shell*:

```
pfc@pfc-escc:~$ echo $SHELL
/bin/bash
pfc@pfc-escc:~$ echo $BASH_VERSION
4.1.5(1)-release
```

El cambio de *Shell* en el sistema se establece sencillo mediante el comando `chsh`.

10.2.2 Nuevos Servicios y Aplicaciones en el Marco. Mejoras Relacionadas con la Arquitectura Funcional

El *Marco* podría establecer diferentes evoluciones. Entre ellas se podrían añadir nuevas mejoras, tales como:

- *Redundancia y Balanceo de Carga.* Una de las características más interesantes, para cualquier sistema de comunicación, es el hecho de poder establecer un sistema con *redundancia* para alta disponibilidad que, incluso, permita establecer un *balanceo de carga* entre las peticiones de configuración y contenido. Para obtener dicho objetivo habría que realizar ciertas variaciones en el *Marco* que afectan a varios de los equipos de la *arquitectura funcional*. Habría que establecer un mecanismo en el que el *ESCC principal* replique la configuración y el contenido sobre los *ESCCs secundarios*. La publicación de estos *ESCC secundarios*, de algún modo, debería ser gestionado por los *ERCs* a la hora de obtener la configuración y sincronización de contenido; por tanto, el *Marco* debe ser modificado también en la parte correspondiente de los *ERCs*, así como en la reproducción del contenido.
- *Equipos Servidores de Contenido Externo.* El contenido multimedia se podría almacenar en servidores independientes. Esto podría ser interesante debido a que archivos multimedia almacenados en alta calidad podrían ocupar una gran cantidad de espacio físico en disco.
- *Servidor de Copia de Seguridad (Backup).* Permitir seleccionar un equipo en el cual volcar una copia de seguridad. Es una herramienta importante a la hora de recuperar fallos de sistema. La configuración que debería permitir el *ESCC*, sería la de seleccionar un servidor *TFTP*, *FTP*, *sFTP* o un dispositivo de almacenamiento masivo (junto con los parámetros necesarios para el acceso: *usuario* y *contraseña*) donde volcar la copia de seguridad del sistema. Una buena práctica será realizar inicialmente un empaquetado de todos los archivos necesarios mediante la llamada a terceras aplicaciones, como puede ser `tar` o `rar`, y, posteriormente, realizar la copia sobre el servidor remoto. Adicionalmente, se debería proveer un mecanismo para que, al cargar la copia de seguridad en el *ESCC*, el sistema se estableciera con dicha configuración.
- *Verificar Disponibilidad del ESCC y sistema de alta disponibilidad.* Implementar un mecanismo de detección de conectividad que permita, bien emitir un contenido explícitamente configurado para estas situaciones o bien la no actualización de peticiones al servidor ante cambios de intervalos de reproducción.

- *Permitir Difusión Selectiva tras redes NAT mediante otras estrategias y parámetros diferenciadores.* En el desarrollo de la aplicación de *Difusión Selectiva*, todas las peticiones atendidas desde una red *nateada*⁴⁰, son atendidas como si tuvieran el mismo origen, debido a que la *dirección IPv4* es el *parámetro diferenciador* escogido. Para añadir esta funcionalidad, una alternativa es enviar, en la propia petición *http*, la *dirección MAC* del *ERC*; sin embargo, la programación del contenido entonces debería estar asociada a la *dirección MAC* y no a la *dirección IPv4*, por lo que debería modificarse tanto el *modelo de datos*, la *Interfaz Gráfica de Configuración* y el modo de gestionar las peticiones en el *ESCC*.
- *Permitir auto-configuración con DHCP.* Una característica interesante encontrada en otros sistemas de *Difusión Selectiva*, es la capacidad de obtener la *dirección IPv4* del equipo mediante el protocolo *DHCP*, lo que permite la comodidad de establecer despliegues masivos sin la necesidad de realizar la configuración de forma explícita equipo por equipo. Sin embargo, en la implementación de un despliegue de este modo se han de tener cuenta diferentes cuestiones:
 - El *parámetro diferenciador* en el sistema de *Difusión Selectiva* debe cambiar o se debe pre-asociar la *dirección MAC* a una *dirección IPV4* en la configuración del servidor *DHCP*.
 - Se ha de idear nuevos mecanismo para obtener la *dirección IPv4* del *ESCC* al cual realizar las peticiones de configuración y contenido; así, como la obtención de ciertos parámetros de configuración del *ERC* (*ancho de banda* para sincronización). Una alternativa muy interesante es la de utilizar las opciones del *protocolo DHCP*[129] [146] para obtener estos parámetros.
- *Contenido en Directo y Difusión Selectiva Mediante Streaming Media.* Se ha incluido la posibilidad de modificar ligeramente el *Marco* para permitir emitir, recibir y reproducir contenido *Streaming Media* mediante flujos *Multicast* y *Unicast*, salvando las limitaciones del *HTTP Streaming* como es la emisión de contenido en directo. Como línea de trabajo futuro se podría proporcionar, a la *Interfaz Gráfica de Usuario*, la posibilidad de programar la recepción de estos flujos que deberán ser accesible desde los *ERCs*.
- *Equipos de Gestión del Flujos Streaming Media en redes WAN.* Se ha valorado, para propagar de forma eficiente los flujos *Streaming Multicast* de contenido multimedia, configurar en los diferentes routers presentes en la red *WAN* los protocolos apropiados y diseñados para tal fin (*IGMP*, *PIM*, *MBGP*); sin embargo, podrían utilizarse otras estrategias totalmente diferentes ante la posibilidad de que el usuario administrador de la aplicación de *Difusión Selectiva*, no tuviera un control absoluto de la red. Así, una de las estrategias consideradas es la de emitir flujos *Unicast* a equipos que se encuentren en sedes remotas a través de la red *WAN* de modo que, estos equipos, sean los encargados de emitir el flujo mediante técnicas *Multicast* en la *LAN* destino. La posibilidad de llevar a cabo esta estrategia se establece de forma sencilla, sin más que utilizar la *aplicación VLC* sobre uno de los *ERCs* de la red destino que adquiera el flujo *Unicast* (*UDP*, *RTP*, *HTTP*, *HTTPS*, *FTP*, *MMS*, *RSTP*) y lo transforme en un flujo *Multicast* (*UDP/RTP Multicast*).
- *Desarrollo y Estudio de la Arquitectura Funcional Proxy Caché.* Como se ha comentado en el Capítulo 4 una alternativa a la *arquitectura funcional desarrollada* (*Auto-Proxy*), es la que se ha definido como *arquitectura funcional proxy-caché* en el contexto de la aplicación de *Cartelería Digital*, así como el modelo híbrido de ésta con la *arquitectura funcional Auto-Proxy*. Una posible línea de trabajo futuro puede ser el desarrollo de

⁴⁰ Término ampliamente utilizado para indicar que tras un Gateway se está implementado el protocolo NAT y, por tanto, las peticiones recibidas desde los equipos tras ese encaminador tienen la misma dirección IP.

estas arquitecturas con el objetivo de realizar un estudio comparativo y verificar de forma cuantitativa cuál de ellas resulta la mejor solución.

- **Algoritmos P2P (peer-to-peer o punto a punto entre iguales) para la Sincronización de Contenido Multimedia.** Un mecanismo apropiado para la distribución del contenido multimedia puede ser a través de algoritmos y protocolos P2P. De esta forma, cada ERC de la red, se comportará como servidor de contenido parcial hacia el resto de ERCs, en vez de establecer la sincronización directamente desde el ESCC. Este mecanismo puede aportar ventajas significativas a la hora de distribuir el contenido multimedia ofreciendo un uso distribuido del *ancho de banda*; por lo que el sistema, con esta estrategia, podrían ser evaluado en una línea de trabajo futuro. En la posible implementación se han de considerar algunos aspectos:
 - En una situación P2P de distribución de contenido, no solo influye el *ancho de banda* de descarga (*download*) de la sede con respecto al ESCC; sino que la velocidad de transferencia depende, tanto de la velocidad de carga (*upload*), como de descarga (*download*), con el resto de sedes y en muchos casos suele ser un mecanismo asimétrico por lo que queda limitada por el valor menor.
 - En determinadas situaciones la propagación del contenido puede ser más lenta, debido a que en cada equipo se debe recibir cada parte de cada archivo.
 - Si los ERCs realizan la descarga de contenido y configuración desde el ESCC, como en el *Marco* desarrollado, tan sólo se debe realizar el encaminamiento desde el ERC a la sede principal donde se encuentra el ESCC. Por el contrario, en una estrategia P2P de descarga de contenido, debe configurarse el encaminamiento en la red entre todas las sedes donde se encuentre un ERC.

10.2.3 Implementación Difusión Selectiva Interactiva

Se ha comentado a lo largo del proyecto fin de carrera, la posibilidad de desarrollar un mecanismo que implemente un sistema de *Difusión Selectiva Interactivo*. Es posible presentar la *Difusión Selectiva* de contenido multimedia de acuerdo a la interacción de los SPC con el *público real*, a través de algún mecanismo de control en los ERCs.

Esta línea de trabajo futuro se puede dividir en varias líneas de trabajo y estudio. Entre ellas algunas que se han identificado son las siguientes:

- Establecer, dentro del *Marco*, la posibilidad de programar y presentar directamente hacia el *público real* páginas web externas (*HTML*) con contenido multimedia embebido que permita la interactividad mediante pantallas táctiles.
- Desarrollar un *servidor socket* y una aplicación que se ejecuten en el ERC para establecer interactividad. La aplicación en el ERC, debe actuar a modo de interfaz entre los sensores que proporcionan información y el *público real*. De esta modo, según la información recogida de los sensores, la aplicación decidirá la orden, de todas las disponibles, que se deberá escribir en el *servidor socket local* de acuerdo con el *Marco* desarrollado. De este modo, la aplicación de control embebida en el navegador que presenta el contenido podrá ajustar las listas de reproducción. Algunos de los sensores podrían estar gobernados por electrónica de uso general o ser equipamiento electrónico como cámaras infrarrojas para el reconocimiento de patrones. Así, entre los sensores, se podría hacer uso de:
 - *Sensores de Temperatura y Humedad*: en función de la sensación térmica y condiciones climáticas se podría variar el contenido multimedia de acuerdo a productos que puedan ser ofrecidos al *público objetivo*.
 - *Sensores de Presencia, Movimiento y Ruidos*: de acuerdo a la presencia o no de *público objetivo* el sistema podría pasar a aumentar la luminosidad

para establecerse más llamativos o ejecutar de forma automática la función de *ahorro de energía*, dependiendo de la presencia o no de público real.

- *Cámaras*: según el reconocimiento, por ejemplo, del sexo del público real se podrá variar el tipo de contenido multimedia. Otras características fisiológicas podría ser el reconocimiento de tallas con el objetivo de presentar ofertas.

10.2.4 Mejorar Aspectos Relacionados con la Seguridad

La seguridad es un aspecto muy importante en un sistema de *Difusión Selectiva* ya que la modificación del contenido, con la inclusión de vídeos no deseados debido a ataques de *crackers*, pueden dilapidar muy rápidamente la imagen de una organización.

Así, se espera que como línea de trabajo futuro se añadan nuevas configuraciones y desarrollos que aporten una mayor seguridad al *Marco* desarrollado:

- *Seguridad en el Acceso a NTP*. Establecer un criterio alternativo en lo concerniente a la configuración de hora de los equipos del sistema; por ejemplo, añadiendo autenticado y utilizando una clave común a todos los *ERCs* en el sistema.
- *Seguridad en la Aplicación rsync*. Establecer un mecanismo de control de acceso a la aplicación *rsync* del *ESCC* desde los *ERCs*.
- *Mecanismo Programado para la Actualización los Sistemas Operativos Utilizados*.
- *Generación Restrictiva del Archivo Crossdomian.xml y Políticas de Acceso Socket*. Generar de forma automática el archivo *crossdomian.xml* y la *política de acceso socket*, para permitir gestionar ciertos accesos a las *aplicaciones swf*, considerando los *ERCs* dados de alta en el sistema.
- *Añadir Encriptado en las Comunicaciones de las aplicaciones swf*. En la implementación desarrollada se ha establecido la comunicación de las peticiones y respuestas mediante peticiones/respuestas *http/XML* con el servidor. Mediante la utilización de nuevas *clases* como las desarrolladas por el proyecto *as3crypto* [147], se puede establecer comunicaciones cifradas.

10.2.5 Instalación Automatizada

Una última línea de trabajo futura es establecer un mecanismo para la instalación y configuración de todo el sistema.

Establecer la instalación del sistema con la llamada a un script que realice las configuraciones descritas en el proyecto es muy sencillo y no entraña ninguna dificultad; no obstante, existen otras herramientas que pueden facilitar un despliegue ante una situación comercial. Uno de los ejemplos más significativo, de acuerdo a la herramientas utilizadas en el desarrollo del *Marco*, lo encontramos en la herramienta *UCK (Ubuntu Customization Kit)* [148], la cual permite generar distribuciones de *Ubuntu* de acuerdo a configuraciones y aplicaciones preestablecidas. Así, se podría generar una distribución para cada uno de los equipos de la *arquitectura funcional* y, de este modo, el potencial usuario de la aplicación podría instalar los diferentes equipos de red, sin más que introducir un *CD* de instalación en el equipo y seguir los típicos pasos de instalación de una distribución *Linux*.

PARTE VI. APÉNDICES

A Presupuesto

A.I. Introducción

En los sucesivos apartados se muestra de forma detallada la viabilidad del proyecto en términos de costes. Se hace una estimación del coste total para la ejecución del proyecto. Para realizar un cálculo en términos de costes para el *Diseño y Desarrollo de un Marco de Aplicaciones de Difusión Selectiva* se ha dividido el proyecto en diferentes **fases** las cuales a su vez se subdividen en **tareas** más específicas, las cuales pueden ser evaluadas debido a que se realizan en un determinado periodo de tiempo concreto. Para el cálculo total del coste del proyecto se realizará un sumatorio del coste de cada tarea individual y le añadiremos el coste de los materiales utilizados durante el desarrollo del proyecto.

A.II. Descomposición en Fases y Tareas

El proyecto se ha organizado en fases. La finalización de cada fase se corresponde con la obtención de un hito temporal el cual es necesario ser alcanzado para la finalización del proyecto. Así, cada fase se ha dividido en varias tareas y existe una dependencia entre éstas dentro de cada fase. La ejecución del conjunto de dichas tareas permite la finalización de cada fase.

En los sucesivos subapartados se hace una breve descripción de cada tarea asociada a cada fase, los objetivos de la misma, las dependencias con las tareas de la misma fase, la duración y el esfuerzo asociado a cada una de ellas.

Para el cálculo del esfuerzo asociado a cada tarea se ha considerado una jornada de trabajo de 8 horas /día y 22 días/mes.

Las fases correspondientes al proyecto fin de carrera se listan a continuación junto con una breve descripción:

- FASE 1: Documentación y Análisis del Estado de Arte
- FASE 2: Diseño del Marco para Aplicaciones de Difusión Selectiva
- FASE 3: Desarrollo de una Interfaz Gráfica de Configuración del Servidor de Contenido. División en Capas
- FASE 4: Desarrollo del Interfaz de Configuración del Equipo Terminal Cliente. División en Capas del Equipo Reproductor Cliente
- FASE 5: Diseño y Desarrollo de un Marco para la Provisión y Visualización de Contenido
- FASE 6: Análisis del Funcionamiento del Marco Diseñado y Desarrollado
- FASE 7: Revisión y Mejora del Marco Desarrollo
- FASE 8: Documentación y Redacción de la Memoria del Proyecto

La notación utilizada para la precedencia de las tareas ha sido la siguiente:

- *Fin a Comienzo (FC)*: La tarea no puede comenzar hasta que finalice la tarea predecesora.
- *Comienzo a Comienzo (CC)*: La tarea no puede comenzar hasta que comience la otra tarea.
- *Fin a Fin (FF)*: La tarea no puede finalizar hasta que finalice la otra tarea.
- *Comienzo a Fin (CF)*: La tarea no puede finalizar hasta que comience la otra tarea.

A.II.1. FASE 1: Documentación y Análisis del Estado de Arte

La importancia de las tareas de la *FASE 1* radica en que en función del estudio realizado en esta fase se diseñará y desarrollará el marco condicionado por las tecnologías y funcionalidades elegidas.

Tarea 1.1: Estudio de las Soluciones Existentes en el Mercado de la Difusión Selectiva

- *Descripción*: En esta tarea se realizará un estudio de las diferentes soluciones presentes en el mercado relacionadas con proporcionar una solución basada en *Difusión Selectiva*. En esta tarea, debido a que la documentación relativa a las soluciones disponibles al inicio del proyecto no se muestra de forma explícita, se realizarán reuniones con proveedores de software y se asistirá a diferentes eventos para conocer en detalle las funcionalidades de los distintos productos.
- *Objetivos*:
 - Evaluar las soluciones de *Difusión Selectiva* presentes en el mercado.
 - Reconocer los sistemas y tecnologías utilizados en las diferentes soluciones evaluadas.
 - Reconocer las funcionalidades de las soluciones evaluadas.
 - Reconocer la arquitectura de las soluciones evaluadas.
 - Reconocer los requerimientos adicionales para implantar las distintas soluciones evaluadas.
- *Dependencias*: Esta tarea comenzará una vez firmado el proyecto.
- *Duración*: 4 semanas.
- *Esfuerzo*: Ingeniero Técnico: 0.7 personas-mes

Tarea 1.2: Estudio en Detallado de la Solución de Referencia

- *Descripción*: En esta tarea se seleccionará una tecnología y arquitectura de referencia, la cual deberá consistir en una solución *Difusión Selectiva* que sea lo más completa posible en términos generales, ya que el objetivo del proyecto es el *Diseño y Desarrollo de un Marco para Aplicaciones de Difusión Selectiva*. Así, se estudiará y documentará en detalle la Arquitectura de Referencia considerando las funcionalidades y los requisitos para implementarla.
- *Objetivos*:
 - Seleccionar una *solución de referencia* no sólo enfocada a la *Cartelería Digital*.
 - Evaluar las funcionalidades de la solución de referencia escogida.
 - Evaluar las necesidades de la arquitectura de referencia seleccionada.
- *Dependencias*: Esta tarea podrá comenzar aproximadamente cuando falte 1 semana para la conclusión de la *Tarea 1.1 (FC-1 sem.)*.
- *Duración*: 2 semanas.
- *Esfuerzo*: Ingeniero: 0.8 personas-mes

Tarea 1.3: Selección de las Características Generales del Sistema a Implementar

- **Descripción:** En esta tarea se persigue definir las características generales del sistema a desarrollar. Se definirán la forma en el que el administrador del sistema accederá a la configuración de éste y las funcionalidades del sistema. Las funcionalidades del sistema estarán basadas en las soluciones existentes en el mercado y la solución escogida como de referencia.
- **Objetivos:**
 - Identificar las alternativas de acceso a la configuración del sistema.
 - Definir y elegir el modo de acceso a la configuración del sistema.
 - Definir las funcionalidades generales del sistema.
- **Dependencias:** Esta tarea podrá comenzar cuando falten dos días para finalizar la Tarea 1.2 (FC-2 días).
- **Duración:** 1 semanas.
- **Esfuerzo:** Ingeniero Técnico: 0.9 personas-mes.

Tarea 1.4: Estudio y Pruebas de las Tecnologías para Implementar un Sistema de Difusión Selectiva

- **Descripción:** En esta tarea se estudiarán las diferentes alternativas para desarrollar el *Marco para Aplicaciones de Difusión Selectiva*. Se evaluará el *sistema operativo* que mejor pudiera adaptarse a las necesidades de la implementación, así como las tecnologías disponibles para desarrollar la forma de acceso a la configuración y la compatibilidad con el *sistema operativo* elegido.
- **Objetivos:**
 - Seleccionar las tecnologías utilizadas para el acceso a la configuración del sistema.
 - Seleccionar las tecnologías utilizadas para distribución y presentación de contenido multimedia en elementos remotos.
 - Instalación y pruebas de las tecnologías seleccionadas en el *sistema operativo* elegido.
 - Evaluar los diferentes problemas presentes en los sistemas operativos seleccionados.
 - Identificar una primera estrategia de desarrollo con las tecnologías seleccionadas en los diferentes sistemas operativos seleccionados.
 - Aprender a instalar y a configurar las distintas tecnologías en las que se implementará el Marco que se desarrollará.
- **Dependencias:** Esta tarea podrá comenzar cuando haya concluido la Tarea 1.3 (FC).
- **Duración:** 2 semanas.
- **Esfuerzo:** Ingeniero: 0.5 personas-mes

A.II.2. FASE 2: Diseño del Marco para Aplicaciones de Difusión Selectiva

En la FASE 2 se establece el diseño del *Marco*. Debido a que la *arquitectura funcional* condiciona el diseño de éste, el funcionamiento de la arquitectura será una tarea de gran peso dentro de la FASE 2 y, por tanto, del diseño del *Marco*.

Tarea 2.1: Diseño de la Arquitectura Funcional para Aplicaciones de Difusión Selectiva

- **Descripción:** En esta tarea se diseñará la *Arquitectura Funcional para Aplicaciones de Difusión Selectiva* sobre la que se diseñará el *Marco*. Dentro del diseño de definirán

elementos dentro de la arquitectura y el rol que desempeñarán (*ESCC, ERC, SPC,...*). Se definirá la nomenclatura para referirnos a los diferentes equipos y dispositivos que intervienen en el sistema final de forma inequívoca.

- **Objetivos:**
 - Definir la nomenclatura utilizada en el proyecto final de carrera.
 - Definir los *espacios de trabajo* de funcionamiento.
 - Definir las opciones de configuración del administrador del sistema.
 - Definir las funcionalidades de los equipos que intervienen en la arquitectura.
 - Definir la interacción entre los equipos.
- **Dependencias:** Esta tarea podrá comenzar cuando haya concluido la FASE 1 (FC).
- **Duración:** 2 semanas.
- **Esfuerzo:** Ingeniero: 0.8 personas-mes

Tarea 2.2: Diseño del Marco para Aplicaciones de Difusión Selectiva

- **Descripción:** En esta tarea se diseñará el *Marco para Aplicaciones de Difusión Selectiva*. Dentro del diseño se estudiarán previamente las alternativas tecnológicas para su desarrollo. El *Marco* diseñado se establecerá de acuerdo a la arquitectura funcional establecida.
- **Objetivos:**
 - Enumerar las funcionalidades de la aplicación final y definir las necesidades del *Marco*.
 - Estudiar las alternativas tecnológicas para el desarrollo del *Marco*.
 - Elegir las tecnologías, incluyendo, *los sistemas operativos* de los equipos para el desarrollo del *Marco*.
 - Diseñar el *Marco*: diferenciación en función del rol de los equipos.
- **Dependencias:** Esta tarea podrá comenzar cuando haya concluido la Tarea 2.1 (FC).
- **Duración:** 2 semanas.
- **Esfuerzo:** Ingeniero: 0.9 personas-mes

Tarea 2.3: Instalación y Configuración del Software para el Desarrollo del Marco para Aplicaciones de Difusión Selectiva

- **Descripción:** En esta tarea se instalarán las herramientas software necesarias para el desarrollo del *Marco* para aplicaciones de *Difusión Selectiva* diseñado.
- **Objetivos:**
 - Instalar *Sistema Operativo* en los equipos.
 - Instalar aplicaciones *software* para el desarrollo de las aplicaciones.
 - Configurar las diferentes aplicaciones
- **Dependencias:** Esta tarea podrá comenzar cuando haya concluido la Tarea 2.2 (FC).
- **Duración:** 0.5 semana.
- **Esfuerzo:** Ingeniero Técnico: 0.8 personas-mes

A.II.3. FASE 3: Desarrollo de una Interfaz Gráfica de Configuración del Servidor de Contenido. División en Capas

En esta Fase se identifican e independizan las distintas capas de desarrollo del *Equipo Servidor de Contenido y Configuración (ESCC)*. Se aplicará el diseño del *Marco* al desarrollo de las aplicaciones en el *ESCC*.

Tarea 3.1: Servidor. Elección de las Distintas Capas de Funcionamiento del Servidor

- **Descripción:** En esta tarea se establecerá la *arquitectura funcional* del ESCC en diferentes capas, para así establecer un desarrollo en paralelo del sistema en cada capa.
- **Objetivos:**
 - Diseñar cada una de las capas del sistema que sirve las peticiones de configuración y contenido.
 - Definir las necesidades de cada una de las capas para establecer independencia en el desarrollo de cada capa.
 - Definir el modo de comunicación de las diferentes capas.
 - Definición de la base preliminar de cada capa (diseño de la *base de datos* y tipos de información almacenada, esquema de directorios, permisos, ...)
- **Dependencias:** Esta tarea podrá comenzar cuando haya finalizado la FASE 2 (FC).
- **Duración:** 2 semana
- **Esfuerzo:** Ingeniero: 0.7 personas-mes

Tarea 3.2: Servidor. Capa 3: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo Servidor

- **Descripción:** En esta tarea se comenzará a implementar la *interfaz gráfica del usuario* administrador del sistema (definidas en la Tarea 3.1) servidor.
- **Objetivos:**
 - Instalar el software necesario para desarrollar la interfaz de administración del sistema.
 - Aprender los lenguajes de programación y las tecnologías específicas para el desarrollo de la interfaz de administrador del sistema (PHP, AS3, XML, MySQL).
 - Realizar simulación de comunicación de cada elemento programado con la Capa 2 del Marco.
- **Dependencias:** Esta tarea comenzará cuando haya concluido la Tarea 3.1 (FC)
- **Duración:** 6 semanas
- **Esfuerzo:** Ingeniero: 0.8 personas-mes

Tarea 3.3: Servidor. Capa 2: Desarrollo de la Capa de Motor PHP del Equipo Servidor

- **Descripción:** En esta tarea se comenzará a desarrollar la capa que interactúa con la Capa 3 y el Capa 1 del Marco. Funcionará a modo de Gateway entre las tecnologías de las capas 3 y 1, recibiendo peticiones de Capa 3 y realizando peticiones a la Capa 1. El resultado obtenido de la Capa 1 es formateado para que pueda ser interpretado por la Capa 3.
- **Objetivos:**
 - Aprender a programar en el lenguaje seleccionado para la Capa 2.
 - Programación de los *scripts* que gestionarán las peticiones asíncronas del sistema.
 - Realizar simulaciones de la interacción con el nivel superior (Capa 3).
 - Realizar simulaciones de la interacción con el nivel inferior (Capa 1).
- **Dependencias:** Esta tarea podrá comenzar cuando haya finalizado la Tarea 3.1 (FC) y podrá finalizar cuando haya finalizado la Tarea 3.2 (FF).
- **Duración:** 4 semanas
- **Esfuerzo:** Ingeniero Técnico: 0.6 personas-mes

Tarea 3.4: Servidor. Capa 1: Desarrollo de la Capa de la Base de Datos

- **Descripción:** En esta tarea se diseñarán las tablas de la *base de datos* almacenada en el sistema según las necesidades de configuración de la *Capa 1* definidas en la Tarea 3.1
- **Objetivos:**
 - Diseño de las tablas y base de datos presentes en el servidor.
 - Creación de las tablas y bases de datos necesarias para el sistema.
 - Simulación de peticiones de acceso, modificación, creación de registros en las tablas según las peticiones que se realizarán desde la Capa 2.
- **Dependencias:** Esta tarea podrá comenzar cuando haya finalizado la Tarea 3.1 (FC) y podrá finalizar cuando haya finalizado la Tarea 3.2 (FF).
- **Duración:** 2 semanas.
- **Esfuerzo:** Ingeniero: 0.8 personas-mes

Tarea 3.5: Servidor. Capa 0: Desarrollo de la Capa de Sistema

- **Descripción:** En esta tarea se desarrollan las configuraciones relativas a la capa del *sistema operativo*. Se desarrollarán los *scripts* necesarios para el correcto funcionamiento de las aplicaciones de *Difusión Selectiva* que se ejecutan dentro del *Marco* desarrollado. En concreto se ejecutan *scripts* que configuran el equipo conforme a los datos almacenados en la tablas de la *base de datos* de la *Capa 1*; por lo que se establece una comunicación con la *Capa 1* que almacena los datos de forma persistente.
- **Objetivos:**
 - Desarrollar los *scripts* que configuran el sistema conforme a los datos almacenados en la *base de datos*.
 - Realizar las configuraciones pertinentes para ejecutar los *scripts* al arranque del sistema.
- **Dependencias:** Esta tarea podrá comenzar cuando haya finalizado la Tarea 3.1 (FC) y podrá finalizar cuando haya finalizado la Tarea 3.2 (FF).
- **Duración:** 2 semanas.
- **Esfuerzo:** Ingeniero: 0.7 personas-mes

Tarea 3.6: Servidor. Integración y Pruebas de Comunicación Entre Capas.

- **Descripción:** En esta tarea se realiza la integración de todas las capas definidas en el equipo servidor.
- **Objetivos:**
 - Integrar la Capa 3 con la Capa 2.
 - Integrar la Capa 2 con la Capa 1.
 - Integrar la Capa 0 con la Capa 1.
 - Probar que el sistema se configura de forma adecuada a través del interfaz de administración del sistema.
- **Dependencias:** Esta tarea podrá empezar cuando hayan concluido las Tareas 3.2 (FC), 3.3 (FC), 3.4 (FC) y 3.5 (FC).
- **Duración:** 2 semanas
- **Esfuerzo:** Ingeniero: 0.5 personas-mes

A.II.4. FASE 4: Desarrollo del Interfaz de Configuración del Equipo Terminal Cliente. División en Capas del Equipo Reproductor Cliente

De igual modo se divide el desarrollo de la aplicación que configura el *ERC* en capas o niveles. Cada capa puede ser desarrollada por diferentes ingenieros trabajando en paralelo. Puede comenzar cuando haya finalizado la FASE 2 (FC).

Tarea 4.1: Cliente. Elección de las Distintas Capas de Funcionamiento del Equipo Terminal Cliente

- *Descripción:* En esta tarea se diseñarán y definirán las capas y el número de niveles del equipo reproductor cliente para su configuración. Así, se definirán según las necesidades de configuración de estos equipos.
- *Objetivos:*
 - Definición del número de capas del equipo terminal.
 - Definición de las características de cada capa.
 - Definición del modo de interactuar de cada capa.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la FASE 2 (FC).
- *Duración:* 1 semanas
- *Esfuerzo:* Ingeniero: 0.5 personas-mes

Tarea 4.2: Cliente. Capa 2: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo Cliente

- *Descripción:* En esta tarea se desarrollará la capa de interfaz de usuario administrador del equipo reproductor cliente. También se desarrollará la interacción con la capa de nivel inferior.
- *Objetivos:*
 - Desarrollar una interfaz de usuario amigable para que el administrador del sistema pueda configurar el equipo de forma remota: Capa 2 del equipo terminal.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 4.1 (FC).
- *Duración:* 2 semanas
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 4.3: Cliente. Capa 1: Desarrollo de la Capa Base de datos

- *Descripción:* En esta tarea se diseñan y crearán las *bases de datos* y las *tablas* que almacenarán la configuración el equipo. Se realizarán pruebas para verificar la posible integración con el resto de capas colindantes.
- *Objetivos:*
 - Diseñar y crear la *base de datos* que almacena la configuración del equipo.
 - Probar la integración con el nivel A y el Nivel C.
- *Dependencias:* : Esta tarea podrá comenzar cuando haya finalizado la Tarea 4.1 (FC) y podrá finalizar cuando haya finalizado la Tarea 4.2 (FF).
- *Duración:* 2 semanas
- *Esfuerzo:* Ingeniero: 0.3 personas-mes

Tarea 4.4: Cliente. Capa 0: Desarrollo de la Capa de Sistema

- *Descripción:* En esta tarea desarrollarán los *scripts* y aplicaciones necesarias para proveer al equipo de la configuración almacenada de forma persistente.
- *Objetivos:*
 - Configurar el *sistema operativo* para permitir a ejecución de *script*
 - Desarrollar los *script* que accedan a la configuración de la *base de datos*.
 - Programar los *script* para adquirir y aplicar la configuración almacenada.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 4.1 (FC) y podrá finalizar cuando haya finalizado la Tarea 4.2 (FF).
- *Duración:* 1 semana
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 4.5: Cliente. Integración y Pruebas de Comunicación entre Capas

- *Descripción:* En esta tarea realizará la integración de las diferentes capas del equipo terminal.
- *Objetivos:*
 - Proporcionar un sistema de comunicación entre las diferentes capas desarrolladas.
 - Realizar pruebas de funcionamiento en la *Interfaz de Configuración del Equipo Cliente*.
- *Dependencias:* Esta tarea comenzará cuando concluya la Tarea 4.3 (FC), Tarea 4.4 (FC) y Tarea 4.2 (FC).
- *Duración:* 1 semana
- *Esfuerzo:* Ingeniero: 0.5 personas-mes

A.II.5. FASE 5: Diseño y Desarrollo de un Marco para la Provisión y Visualización de Contenido Multimedia

En la FASE 5 se debe diseñar y desarrollar el *Marco* que permita el funcionamiento de la arquitectura atendiendo a la configuración establecida en el sistema. Así, en esta fase se desarrollan las *aplicaciones* y *script* necesarios para presentar el contenido multimedia. Esta fase no debería comenzar hasta se hayan finalizado las fases que permiten introducir la configuración del sistema a través de una interfaz gráfica: FASE 3 (FC) y FASE 4(FC).

Tarea 5.1: Servidor y Cliente. Diseño de un Marco para Aplicaciones de Difusión Selectiva de Contenido Multimedia

- *Descripción:* En esta tarea se diseñará cómo interactuarán el servidor y clientes para que estos últimos obtengan la configuración y los archivos del servidor. También se diseñará como los clientes presentarán el contenido y cómo el servidor reconocerá qué configuración debe proporcionar.
- *Objetivos:*
 - Diseñar en el servidor cómo discernir entre la configuración que debe proporcionar a cada cliente, además de diseñar el modo de comunicación de la configuración.
 - Diseñar cómo el cliente obtendrá la configuración del servidor y cómo presentará el contenido multimedia finalmente.
 - Diseñar un sistema que permita a los clientes reconocer cambios de configuración en el servidor y ajustar el contenido de presentación a los cambios detectados.

- *Dependencias:* Esta tarea comenzará cuando hayan finalizado las FASES 3 (FC) y 4 (FC).
- *Duración:* 1 semana
- *Esfuerzo:* Ingeniero: 0.7 persona-mes

Tarea 5.2: Servidor. Desarrollar un script en el Servidor que Atienda las Peticiones de Configuración de los Equipos Terminales Clientes

- *Descripción:* En esta tarea se desarrollará un *script* que atienda las peticiones de configuración de los equipos terminales. En función de los parámetros considerados, para diferenciar la configuración, se deberá reenviar al equipo terminal la configuración.
- *Objetivos:*
 - Desarrollar *script* que atienda las peticiones de los Clientes
 - Desarrollar *script* que envíe la configuración a los clientes.
- *Dependencias:* Esta tarea comenzará cuando concluya la tarea 5.1 (FC).
- *Duración:* 1 semana
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 5.3: Cliente. Desarrollar el Mecanismo por el cual el Equipo Terminal Cliente Presentará el Contenido Multimedia

- *Descripción:* En esta tarea se desarrollará un sistema por el cual el equipo terminal cliente presentará el contenido multimedia a través de pantallas u otros mecanismos de presentación del contenido hacia la audiencia.
- *Objetivos:*
 - Desarrollar una aplicación que permita reproducir el contenido multimedia.
 - Presentar el contenido de acuerdo a la información recibida de la configuración del servidor (Tarea 5.2)
 - Adaptación del *sistema operativo* del equipo terminal para presentar el contenido directamente al inicio del sistema.
- *Dependencias:* Esta tarea comenzará cuando finalice la Tarea 5.2 (FC).
- *Duración:* 3 semanas
- *Esfuerzo:* Ingeniero: 0.8 personas-mes

Tarea 5.4: Servidor y Cliente. Desarrollar el Mecanismo que Permite al Cliente Detectar Cambios en la Configuración del Contenido Multimedia a Mostrar

- *Descripción:* En esta tarea se desarrollará un sistema de comunicación en el cual el equipo servidor comunicará, a los equipos terminales clientes, las modificaciones establecidas en la configuración del servidor.
- *Objetivos:*
 - Desarrollar un sistema de envío de información desde el equipo servidor a los equipos terminales clientes.
 - Desarrollar un sistema de interpretación de mensajes recibidos desde el equipo servidor y ejecución de acciones en los Clientes
- *Dependencias:* Esta tarea comenzará cuando finalice la Tarea 5.3 (FC).
- *Duración:* 3 semanas
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 5.5: Servidor y Cliente. Verificar el Funcionamiento del Marco Diseñado y Desarrollado

- **Descripción:** En esta tarea se realizarán pruebas de funcionamiento del sistema de comunicación desarrollado entre el Servidor y el Cliente y se comprobará en equipo terminal cliente que la presentación del contenido es la adecuada atendiendo a la configuración establecida
- **Objetivos:**
 - Verificar el sistema de comunicación desarrollado entre equipo servidor y el equipo terminal cliente.
 - Verificar la correcta presentación de contenido multimedia en el medio de presentación.
- **Dependencias:** Esta tarea comenzará cuando haya concluido la Tarea 5.4 (FC)
- **Duración:** 1 semana
- **Esfuerzo:** Ingeniero: 0.5 personas-mes

A.II.6. FASE 6: Análisis del Funcionamiento del Marco Diseñado y Desarrollado

La FASE 6 podrá comenzar cuando se hayan finalizado las FASES 3 (FC), FASE 4 (FC) y FASE 5 (FC).

Tarea 6.1: Instalación del Sistema Servidor y Sistema Cliente en Máquina Virtual

- **Descripción:** En esta tarea se instalará un servidor de máquinas virtuales y se instarán máquinas virtuales con el *sistema operativo* escogido para cada tipo de equipo (servidor y cliente). Se configurarán estos sistemas con los *script* y software desarrollado.
- **Objetivos:**
 - Instalar servidor de máquinas virtuales
 - Aprender los conocimientos básicos para implantar equipos virtualizados.
 - Crear máquina virtual del equipo servidor: instalar el *sistema operativo* e instalar y configurarle el software necesario, cargando el software desarrollado.
 - Crear máquina virtual Cliente: instalar el *sistema operativo*, instalarle y configurar el software necesario y cargarle el software desarrollado.
- **Dependencias:** Esta tarea podrá comenzar cuando hayan finalizado las FASES 3,4 y 5 (FC).
- **Duración:** 1 semana
- **Esfuerzo:** Ingeniero Técnico: 0.5 personas-mes

Tarea 6.2: Instalación de los Equipos de Red para Simular un Ambiente de Producción Real

- **Descripción:** En esta tarea se aprenderá configurar los equipos de red disponibles: *switches*, *routers* y puntos de acceso *wi-fi*. Se instalará el software apropiado a los diferentes equipos y se configurarán de acuerdo a un escenario de pruebas genérico para testar el sistema.
- **Objetivos:**
 - Aprendizaje de la configuración de equipos de red disponibles.
 - Instalación de software apropiado a los distintos equipos de red.

- Configuración de los equipos de red.
- Instalación de software para verificar el funcionamiento de forma cuantitativa.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 6.1 (FC).
- *Duración:* 2 semanas
- *Esfuerzo:* Ingeniero: 0.9 *personas-mes*

Tarea 6.3: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Simple

- *Descripción:* En esta tarea se configurará el Marco Desarrollado y se ejecutará su funcionamiento sobre un Espacio de Trabajo Simple.
- *Objetivos:*
 - Configuración del Servidor.
 - Configuración de los diferentes equipos terminales clientes.
 - Corrección de posibles errores.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 6.2 (FC).
- *Duración:* 3 semana
- *Esfuerzo:* Ingeniero : 0.4 *personas-mes*

Tarea 6.4: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Distribuido

- *Descripción:* En esta tarea se evaluará de forma cualitativa y cuantitativa el *Marco* desarrollado en un espacio de trabajo distribuido..
- *Objetivos:*
 - Evaluación y análisis de resultados en función de la variación de parámetros de red sobre la que funciona el *Marco* desarrollado.
 - Identificar posibles mejoras al *Marco* desarrollado en función de los resultados.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 6.2 (FC) y la Tarea 6.3 (FC).
- *Duración:* 3 semanas
- *Esfuerzo:* Ingeniero: 0.6 *personas-mes*

A.II.7. FASE 7: Revisión y Mejora del Marco Desarrollo

En esta fase se implementarán algunas mejoras encontradas una vez se ha evaluado el funcionamiento del *Marco*. Esta fase no podrá comenzar hasta que no haya finalizado la FASE 6 (FC).

Tarea 7.1: Evaluación de la Viabilidad de Posibles Mejoras del Marco Desarrollado

- *Descripción:* En esta tarea se determinarán las mejoras posibles y su viabilidad en la implementación del Marco para Aplicaciones de *Difusión Selectiva* Desarrollado.
- *Objetivos:*
 - Evaluar la viabilidad las posibles mejoras en la implementación de la *Marco*.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la FASE 6.
- *Duración:* 2 días
- *Esfuerzo:* Ingeniero: 0.8 *personas-mes*

Tarea 7.2: Desarrollo de las Mejoras Seleccionadas

- *Descripción:* En esta tarea se desarrollarán las modificaciones sobre el software para implementar las mejoras consideradas en la Tarea 7.1.
- *Objetivos:*
 - Desarrollar las mejoras identificadas.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 7.1 (FC).
- *Duración:* 2 semanas
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 7.3: Realización de Estudio Comparativo con Mejoras Introducidas en el Marco

- *Descripción:* En esta tarea se estudiará de forma comparativa el Marco finalmente desarrollado con el desarrollado antes de introducir las mejoras.
- *Objetivos:*
 - Evaluación, a nivel cualitativo y cuantitativo, las prestaciones del sistema final.
 - Comparación del sistema final con el anterior.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la Tarea 7.2 (FC).
- *Duración:* 1 semanas
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

A.II.8. FASE 8: Documentación y Redacción de la Memoria del Proyecto

En esta se documentará el trabajo realizado en el desarrollo del proyecto, así como las líneas futuras de trabajo encontradas.

Tarea 8.1: Documentación del Desarrollo del Proyecto y las Pruebas Realizadas

- *Descripción:* En esta tarea se realizará la documentación de todo el proceso de diseño, desarrollo, configuración y despliegue del entorno de pruebas. También se documentarán los resultados de las pruebas obtenidas del *Marco* para aplicaciones *Difusión Selectiva* finalmente implementado.
- *Objetivos:*
 - Documentación de los antecedentes en la tecnología: *Estado del Arte*.
 - Documentación del diseño y configuraciones del *Marco* implementado.
 - Redacción de un informe de pruebas describiendo los diferentes espacios de trabajo configurados así como las pruebas a realizadas y el resultado de éstas.
- *Dependencias:* Esta tarea podrá comenzar cuando haya finalizado la FASE 5 (FC) y podrá finalizar cuando haya finalizado la FASE 7 (FF).
- *Duración:* 4 semanas
- *Esfuerzo:* Ingeniero: 0.7 personas-mes

Tarea 8.2: Redacción del Documento Final de Proyecto

- *Descripción:* En esta tarea se redactará la memoria final del proyecto, comentando los aspectos más interesantes de la realización del mismo. También se incluirá anexos con información destacable y se utilizarán las notas y documentos generados durante la realización del proyecto.
- *Objetivos:*
 - Redacción del a memoria del proyecto final de carrera.
- *Dependencias:* Esta tarea podrá comenzar cuando haya concluido la Tarea 8.1 (FC).

- *Duración: 6 semanas*
- *Esfuerzo: Ingeniero: 0.9 personas-mes*

A.III. Resumen del Proyecto

A continuación (Tabla 37), se muestra un resumen de los tiempos empleados en cada tarea:

Tareas	Duración	Esfuerzo	Total
Documentación y Análisis del Estado del Arte			
Tarea 1.1: Estudio de las Soluciones Existentes en el Mercado de la Difusión Selectiva	4 sem.	0.7 h-m	120 h
Ingeniero Técnico			
Tarea 1.2: Estudio en Detallado de la Solución de Referencia	2 sem.	0.8 h-m	70 h
Ingeniero			
Tarea 1.3: Selección de las Características Generales del Sistema a Implementar	1 sem.	0.9 h-m	40 h
Ingeniero			
Tarea 1.4: Estudio y Pruebas de las Tecnologías para Implementar un Sistema de Difusión Selectiva	2 sem.	0.5 h-m	40 h
Ingeniero			
Total Fase:	270 h		
Diseño del Marco para Aplicaciones de Difusión Selectiva			
Tarea 2.1: Diseño de la Arquitectura Funcional para Aplicaciones de Difusión Selectiva	2 sem.	0.8 h-m	66 h
Ingeniero			
Tarea 2.2: Diseño del Marco para Aplicaciones de Difusión Selectiva	2 sem.	0.9 h-m	80 h
Ingeniero			
Tarea 2.3: Instalación y Configuración del Software para el Desarrollo del Marco para Aplicaciones de Difusión Selectiva	0.5 sem.	0.8 h-m	18 h
Ingeniero Técnico			
Total Fase:	164 h		
Desarrollo de una Interfaz Gráfica de Configuración del Servidor de Contenido			
Tarea 3.1: Servidor. Elección de las Distintas Capas de Funcionamiento del Servidor	2 sem.	0.7 h-m	60 h
Ingeniero			
Tarea 3.2: Servidor. Capa 3: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo Servidor	6 sem.	0.8 h-m	200 h
Ingeniero			
Tarea 3.3: Servidor. Capa 2: Desarrollo de la Capa de Motor PHP del Equipo Servidor	4 sem.	0.6 h-m	100 h
Ingeniero			
Tarea 3.4: Servidor. Capa 1: Desarrollo de la Capa de la Base de Datos	2 sem.	0.8 h-m	70 h
Ingeniero			
Tarea 3.5: Servidor. Capa 0: Desarrollo de la Capa de Sistema	2 sem.	0.7 h-m	60 h
Ingeniero			
Tarea 3.6: Servidor. Integración y Pruebas de Comunicación Entre Capas.	2 sem.	0.5 h-m	44 h
Ingeniero			
Total Fase:	534 h		
Desarrollo de una Interfaz Gráfica de Configuración del Equipo Terminal			
Tarea 4.1: Cliente. Elección de las Distintas Capas de Funcionamiento del Equipo	1 sem.	0.5 h-m	20 h
Ingeniero			
Tarea 4.2: Cliente. Capa 2: Desarrollo de la Capa Interfaz de Usuario Administrador del Equipo Cliente	2 sem.	0,7 h-m	70 h
Ingeniero			
Tarea 4.3: Cliente. Capa 1: Desarrollo de la Capa Base de datos	2 sem.	0.3 h-m	30 h
Ingeniero			
Tarea 4.4: Cliente. Capa 0: Desarrollo de la Capa de Sistema	1 sem.	0.7 h-m	30 h
Ingeniero			
Tarea 4.5: Cliente. Integración y Pruebas de Comunicación entre Capas	1 sem.	0.5 h-m	20 h
Ingeniero			
Total Fase:	170 h		

Tareas	Duración	Esfuerzo	Total
Diseño y Desarrollo de un Marco para la provisión y Visualización de Contenido Multimedia mediante Difusión Selectiva			
Tarea 5.1: Servidor y Cliente. Diseño de un Marco para Aplicaciones de Difusión Selectiva de Contenido Multimedia	1 sem.	0.7 h-m	30 h
Ingeniero			
Tarea 5.2: Servidor. Desarrollar un script en el Servidor que Atienda las Peticiones de Configuración de los Equipos Terminales Clientes	1 sem.	0.7 h-m	30 h
Ingeniero			
Tarea 5.3: Cliente. Desarrollar el Mecanismo por el cual el Equipo Terminal Cliente Presentará el Contenido Multimedia	3 sem.	0.8 h-m	100 h
Ingeniero			
Tarea 5.4: Servidor y Cliente. Desarrollar el Mecanismo que Permite al Cliente Detectar Cambios en la Configuración del Contenido Multimedia a Mostrar	3 sem.	0.7 h-m	90 h
Ingeniero			
Tarea 5.5: Servidor y Cliente. Verificar el Funcionamiento del Marco Diseñado y Desarrollado	1 sem.	0.5 h-m	20 h
Ingeniero			
Total Fase:			270 h
Análisis de Funcionamiento del Marco Diseñado y Desarrollado			
Tarea 6.1: Instalación del Sistema Servidor y Sistema Cliente en Máquina Virtual	1 sem.	0.5 h-m	20 h
Ingeniero Técnico			
Tarea 6.2: Instalación de los Equipos de Red para Simular un Ambiente de Producción Real	2 sem.	0.9 h-m	80 h
Ingeniero			
Tarea 6.3: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Simple	3 sem.	0.4 h-m	50 h
Ingeniero			
Tarea 6.4: Realización de Pruebas del Marco Desarrollado en un Espacio de Trabajo Distribuido	3 sem.	0.6 h-m	80 h
Ingeniero			
Total Fase:			230 h
Revisión y Mejora del Marco Desarrollado			
Tarea 7.1: Evaluación de la Viabilidad de Posibles Mejoras del Marco Desarrollado	3 d	0.8 h-m	20 h
Ingeniero			
Tarea 7.2: Desarrollo de las Mejoras Seleccionadas	1 sem.	0.7 h-m	30 h
Ingeniero			
Tarea 7.3: Realización de Estudio Comparativo con Mejoras Introducidas en el	1 sem.	0.7 h-m	30 h
Ingeniero			
Total Fase:			80 h
Documentación y Redacción de la Memoria del Proyecto			
Tarea 8.1: Documentación del Desarrollo del Proyecto y las Pruebas	4 sem.	0.7 h-m	140 h
Ingeniero			
Tarea 8.2: Redacción del Documento Final de Proyecto	6 sem.	0.9 h-m	240 h
Ingeniero			
Total Fase:			380 h
TOTAL PROYECTO			
Total Proyecto:			2.098 h

Tabla 37.-. Resumen del Proyecto.

A.IV. Diagrama de Gantt

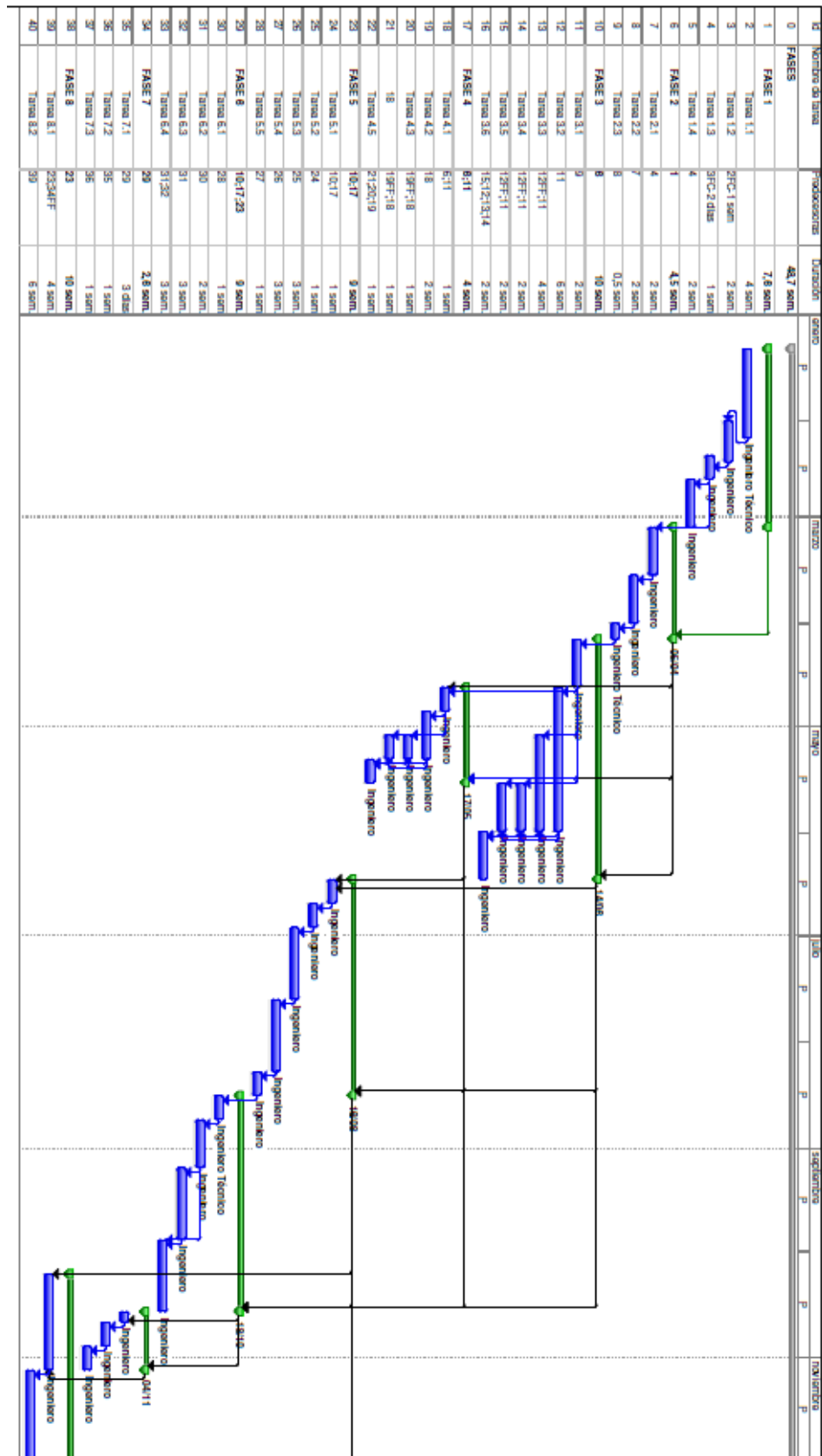


Ilustración 91.- Diagrama de Gantt del Proyecto.

A.V. Costes del Proyecto

En este apartado se realiza un resumen desglosado de los costes totales del proyecto.

Siguiendo las directivas europeas (*Directiva Bolkestein, Directiva 2006/123/CE*) se publicó en el *BOE (Boletín Oficial del Estado)* la Ley 25/2009, de 22 de diciembre, la cual modifica diversas leyes para su adaptación a la Ley sobre el libre acceso a las actividades de servicios y su ejercicio. Esta ley es conocida como *Ley Ómnibus*. Como aspecto reseñable de esta ley, y que afecta al cálculo del coste del proyecto, se tiene que la Ley 2/1974, de 13 de febrero, sobre *Colegios Profesionales*, queda modificada en los siguientes términos:

«Artículo 14. Prohibición de recomendaciones sobre honorarios.

Los Colegios Profesionales y sus organizaciones colegiales no podrán establecer baremos orientativos ni cualquier otra orientación, recomendación, directriz, norma o regla sobre honorarios profesionales, salvo lo establecido en la Disposición adicional cuarta.»

Debido a que la disposición cuarta, a la que se refiere el artículo, es sobre las costas jurídicas en caso de litigio, los datos de los costes asociados de la hora de un *Ingeniero Superior de Telecomunicación* y de un *Ingeniero Técnico de Telecomunicación*, en la actualidad no pueden establecerse con una referencia oficial de un *Colegio Profesional*. Así, el COIT (*Colegio Oficial de Ingenieros de Telecomunicación*) estableció las últimas referencias en el año 2008 las cuales han sido tomadas para establecer un criterio:

Trabajador	COIT 2008		COSTE	
Ingeniero Superior de Telecomunicación	Hora Normal:	78€	Hora Normal:	80€
	Hora Extra:	99€	Hora Extra:	105€
Ingeniero Técnico de Telecomunicación	Hora Normal:	47€	Hora Normal:	50€
	Hora Extra:	60€	Hora Extra:	66€

Tabla 38.-. Referencia tomada para el cálculo de los costes de personal.

Desglosando los costes del proyecto, se tiene:

- **Coste de Personal** **163.700 €**
 - Ingeniero Superior de Telecomunicaciones:

$1.960\text{ h } (2.098\text{ h} - 138\text{ h}) \times 80\text{ €/horas} = 156.800\text{ €}$
 - Ingeniero Técnico de Telecomunicación:

$138\text{ h} \times 50\text{ €/horas} = 6.900\text{ €}$
- **Costes de Material** **4.990 €**
 - **Hardware** **4.040 €**
 - Router (QoS, dot1q) 2 unidades x 800 € 1.600 €
 - Switch (VLAN, dot1q) 1 unidad x 300 € 300 €
 - Access Point (DD-WRT) 1 unidad x 100 € 100 €
 - Ordenadores portátiles 2 unidades x 1000 € = 2.000 €
 - Cables Varios 40 €
 - **Software** 950 €

▪	SO Microsoft Windows XP	1 licencia	300€
▪	IOS Routers	bundle	0 €
▪	Adobe Flash Professional	1 licencia	600 €
▪	Origin	1 licencia (student)	50 €
•	Impuestos (18%)		30.364,2 €
•	TOTAL PROYECTO		194.064,2 €

El coste total del proyecto asciende a *ciento noventa y cuatro mil sesenta y cuatro euros con veinte céntimos de euro*.

Leganés, 12 de abril de 2011

El Ingeniero Projectista

Fdo. David Timón Morillo-Velarde

B Instalación y Configuración del Entorno de Pruebas

B.I. Instalación de Servidor de Máquinas Virtuales. VMWare Server

Para la realización de pruebas y test del *Marco* desarrollado se ha requerido la instalación de la aplicación *VMWare Server* [55]. Mediante esta aplicación es posible emular la instalación de los sistemas sobre arquitecturas x86. La instalación de la aplicación VMWare es muy sencilla y sólo hay que seguir los pasos de instalación en el *sistema operativo* es Windows XP [95]. En otros sistemas operativos como Linux, según en qué versiones se han de ajustar algunos parámetros. La versión de este software utilizada para realizar las pruebas de funcionamiento del *Marco* desarrollado ha sido la 2.0.2.

Para asignar nuevas interfaces virtuales a cada sistema virtualizado se han de realizar algunas configuraciones en el sistema. En principio se ha de asociar a cada interfaz virtual una interfaz física. De este modo al seleccionar en la aplicación VMWare del *equipo anfitrión* una interfaz virtual se establece el flujo de datos por la interfaz asociada:

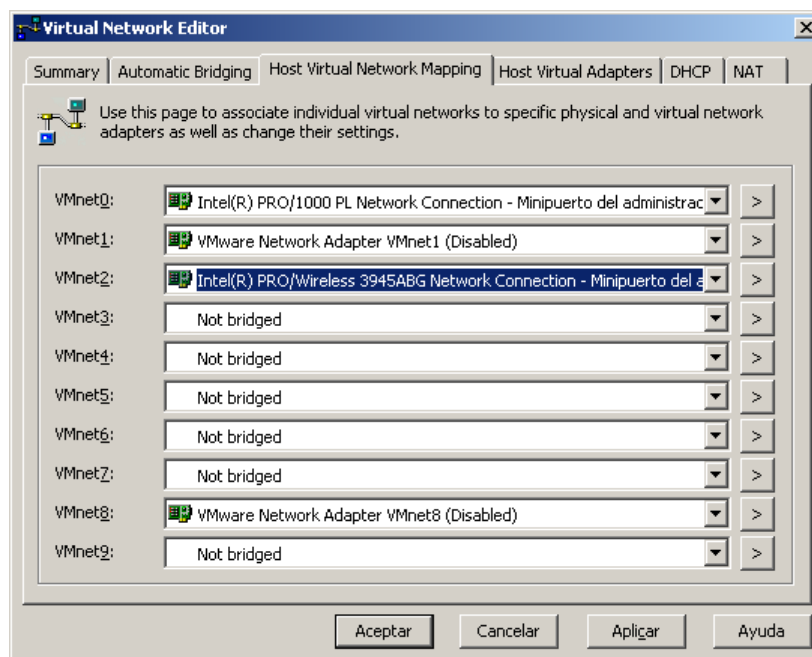


Ilustración 92.-. Editor de Red Virtual de la aplicación VMWare Server. Asociación de interfaz virtual e interfaz física.

Tras la asociación de las interfaces se ha de habilitar, en la interfaz física, la capacidad para permitir a las máquinas virtuales utilizar dicha interfaz física. Se establece a través del marcado de *VMWare Bridge Protocol* en las interfaces que se vayan a utilizar:

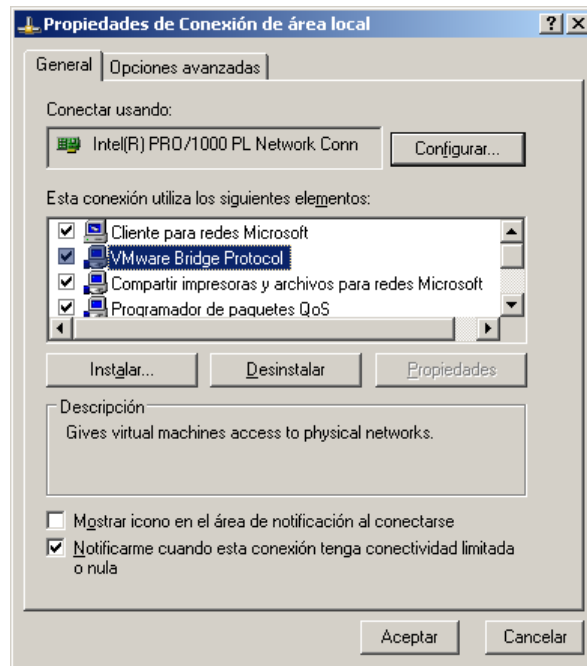


Ilustración 93.- Configuración para proporcionar a las máquinas virtuales la capacidad de utilizar las interfaces físicas.

Posteriormente, al hardware de la máquina virtual se ha de asociar, al menos, una interfaz virtual para establecer la conectividad en la red física.

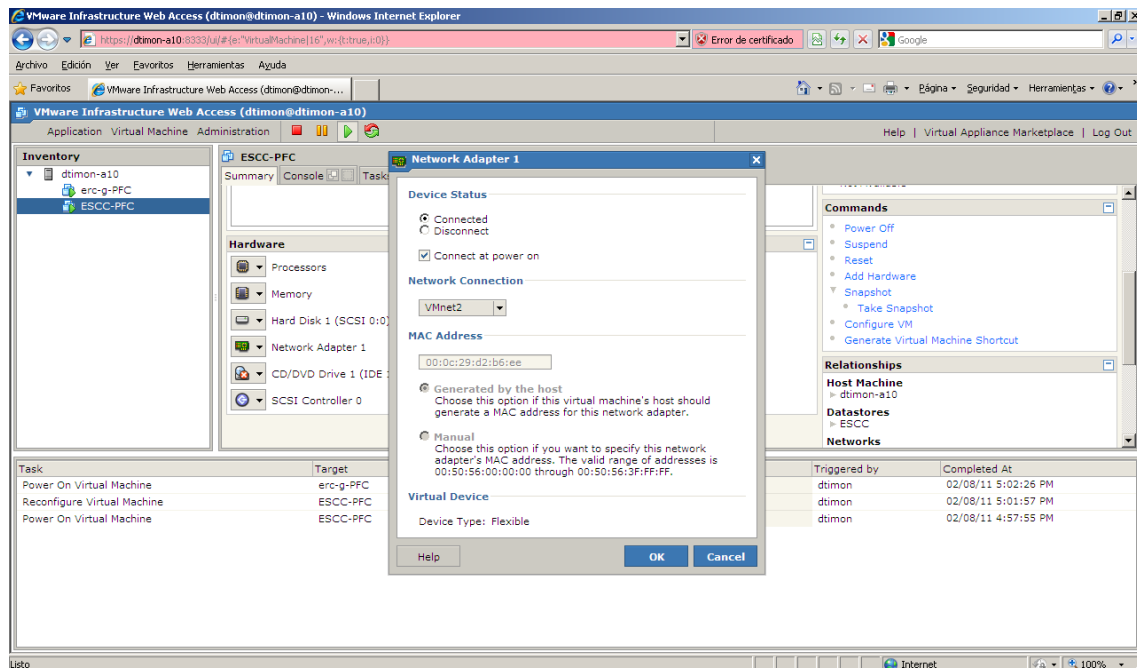


Ilustración 94.- Configuración del hardware virtual sobre la asociación de las interfaces definida.

Finalmente, para mantener conectividad desde el *equipo anfitrión* hacia las máquinas virtuales, se deberán configurar el interfaz físico compartido en la misma subred que la configuración de la máquina virtual.

B.II. Configuración BW WAN en Espacio Distribuido con Varias Sedes

El software de los routers utilizados dispone de varios mecanismos para limitar el *ancho de banda*; por ejemplo, mediante colas y *conformado de tráfico*, así como otros mecanismos más sofisticados⁴¹. Sin embargo, para el propósito del proyecto, se hace uso de la configuración de la *Tasa de Acceso Comprometida* (CAR, *Committed Access Rate*) y se aplica una *política de calidad de servicio* (QoS): paquetes *transmitidos* o *descartados*, en función de si se supera una determinada *Tasa de Acceso* (CAR o *Tasa de Acceso Comprometida*).

En la siguiente ilustración se muestra la diferencia principal entre el *conformado de tráfico* y el establecimiento de una *política*:

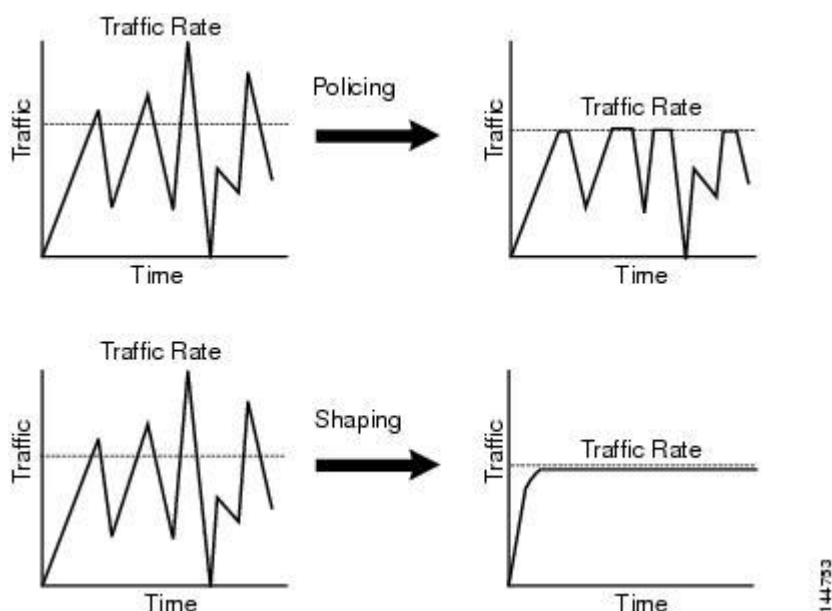


Ilustración 95.-. Diferencias entre conformado de tráfico y aplicación de política de calidad de servicio.
Fuente Cisco 2010.

Con la aplicación de una *política* a una CAR es posible limitar la velocidad de entrada o salida para un tipo de tráfico determinado y sobre un interface o sub interface concreto. Así, con una configuración apropiada se permite descartar el tráfico que excede de la velocidad máxima establecida, por lo que es un mecanismo apropiado para el propósito de las pruebas realizadas en el proyecto a la hora de limitar el *ancho de banda* de cada sede.

La configuración que se va a establecer en el router funciona exclusivamente con las versiones de IOS que soportan QoS. En el contexto del *proyecto final de carrera* se establecerá, por tanto, en el Router A, el cual dispone de una IOS capaz de soportar QoS: c1700-y7-mz.124-15.T13.bin. En la configuración se limita el *ancho de banda* de los servicios que están 'detrás' (subredes) del Router A en el caso de las pruebas en un *Espacio de Trabajo*

⁴¹ Como *Class-Based Policing* (RFC 2697 y RFC 2698).

Distribuido. En nuestro caso, el servicio dependerá del *direccionamiento IPv4* en cada sub interfaz configurada.

El esquema del funcionamiento del algoritmo se muestra en la siguiente ilustración:

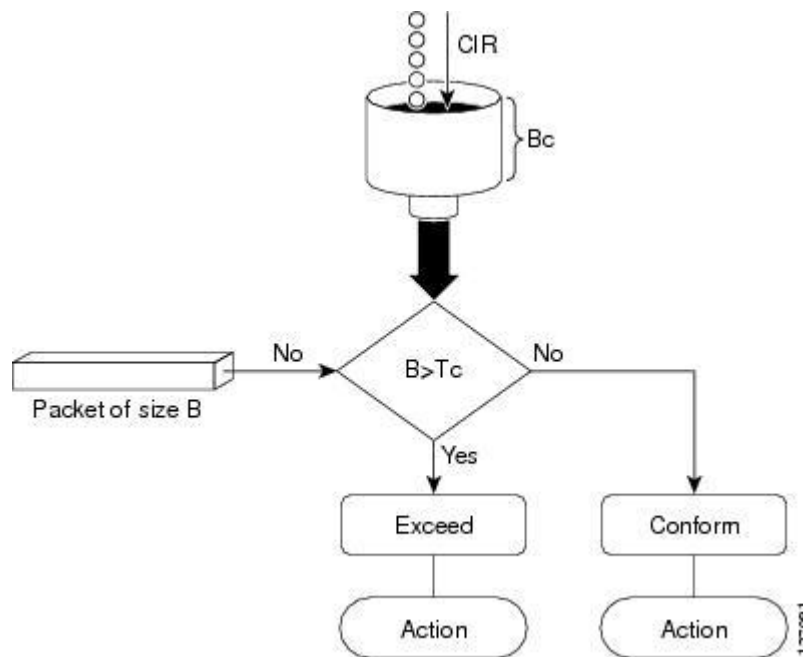


Ilustración 96.-. Esquema del algoritmo Token Bucket. Fuente Cisco 2010.

Se va a utilizar, por tanto, una política para limitar el tráfico basada en el algoritmo *Token Bucket*.

Hay tres parámetros principales o fundamentales que aplican al algoritmo más simple (un solo cubo de fichas): un *Tamaño de Ráfaga Comprometida* (B_c), una *Tasa Media Comprometida* (CIR) y un *intervalo de tiempo* (T_c). Con cualquiera de dos de los valores se puede deducir el tercero a partir de la siguiente expresión:

$$CIR (Tasa Media) = \frac{B_c}{T_c}$$

Estos términos son definidos como siguen:

- *Tasa Media Comprometida* (CIR , *Committed Information Rate*): Especifica cuantos datos pueden ser enviados en media por unidad de tiempo.
- *Tamaño de la Ráfaga Comprometida* (B_c , *Committed Burst*): Especifica, por cada ráfaga, cuantos datos pueden ser enviados dentro de un tiempo dado.
- *Intervalo de Tiempo en Ráfaga Comprometida* (T_c , *Committed Time*): especifica la cantidad de tiempo en segundos en enviar una ráfaga.

En la metáfora del *cubo de fichas* (*Token Bucket*), las *fichas* (*tokens*) se ponen en el cubo (*bucket*) a una cierta velocidad (CIR). Estas *fichas* representan permisos para que el emisor pueda enviar un cierto número de *bytes* a la red. Para enviar un paquete, el sistema que regula el algoritmo, debe eliminar una serie de fichas del cubo que es igual al tamaño del paquete.

El algoritmo que se aplica a cada sub interfaz del Router A es un algoritmo de *Cubos con Fichas* (algoritmo *Token-Bucket*) en el que se pueden configurar dos ráfagas: una que se enmarcaría dentro del tráfico conforme (*'Conform'*) y la otra etiquetada como tráfico excedido (*'Exceed'*). A cada tipo de tráfico, en función de una *Lista de Acceso* (ACL, *Access List*) y en cada sub interfaz, se le podría aplicar una *política* diferente.

En la siguiente ilustración se muestra el algoritmo *Token Bucket* que se implementa:

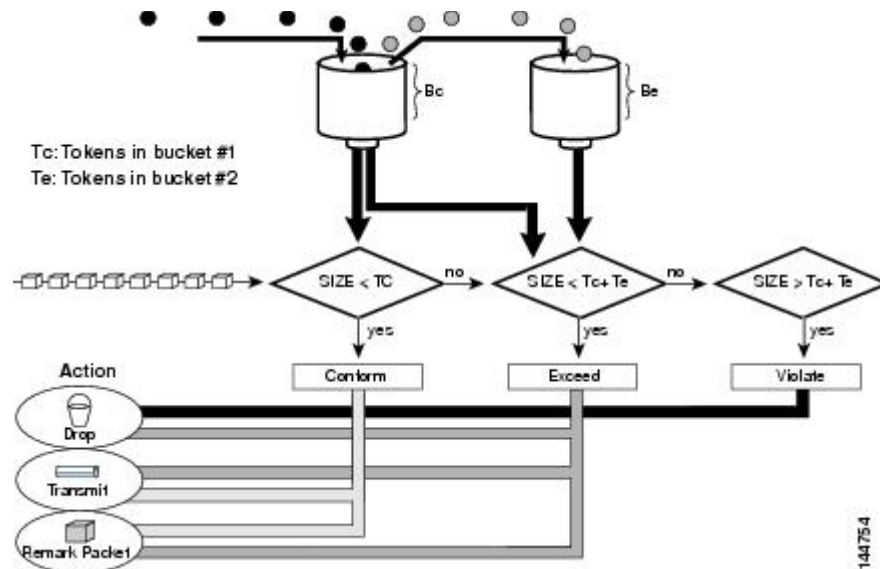


Ilustración 97.- Algoritmo de 2 Cubos con Fichas (Token Bucket). Fuente Cisco 2010.

La Ilustración 97 representa el algoritmo *Token-Bucket* con dos cubos el cual es el algoritmo aplicado por la *IOS* del router en el proyecto fin de carrera; el primer cubo es el cubo de tráfico conforme (*'Conform'*) y el segundo cubo es el cubo tráfico excedido o sobrepasado (*'Exceed'*).

- El tamaño completo del *cubo de tráfico conforme* es el número de bytes especificado como el tamaño de ráfaga normal (*Bc*).
- El tamaño total del *cubo de tráfico excedido* es el número de bytes especificado en el tamaño de ráfaga máxima (*Be*).

Ambos cubos están inicialmente llenos y son actualizados en base a la tasa con la que llegan las *fichas*, la cual es determinada por el *CIR*.

Si el número de *bytes* del paquete que llega para ser transmitido es menor que el número de *bytes* en el *cubo de tráfico conforme*, el paquete se considera conforme. El número necesario de fichas, para el paquete que ha llegado, se eliminan del *cubo de tráfico conforme* y se toman las medidas que se hayan configurado por la *política* (por ejemplo, el paquete se transmite).

Si en el *cubo de tráfico conforme* no contiene fichas suficientes, se comprueban las fichas del *cubo de tráfico excedido* con el número de *bytes* del paquete que se pretende transmitir. Si la combinación de las fichas presentes de ambos cubos es superior a la que se necesita para enviar el paquete se ejecuta la configuración *Exceed* y se retiran el número de fichas necesarias de cada cubo. Si el *cubo de tráfico excedido* contiene un número insuficiente de fichas para enviar esos bytes, el paquete se considera con el estado *Violate* y el paquete es descartado.

Para configurar el *ancho de banda* en cada una de las distintas sedes emuladas se han configurado *Listas de Acceso (ACL, Access List)* en el router principal (*Router A*), las cuales se aplican en las diferentes sub interfaces utilizadas a modo de *política*. El tráfico permitido en cada sub interfaz, y por tanto cada subred (ver Ilustración 61 e Ilustración 62), se configura en función de la configuración de la propia interfaz. En cada sub interfaz se puede limitar tanto el tráfico entrante (*BW-UL; UpLoad* desde el *ERC* al *ESCC*), como el saliente (*BW-DL; DownLoad* desde el *ESCC* al *ERC*) con una configuración diferente; sin embargo, se establece un tráfico simétrico (ver apéndice B.III.1 Router A).

Como se ha observado en las diferentes pruebas, el tráfico importante y flujo de *ancho de banda* considerable se establece desde el *ESCC* a los *ERC* ya que desde el servidor se transmiten los *archivos multimedia* a difundir. En las pruebas realizadas no se han establecido configuración de *enlaces asimétricos* por no considerarse de especial interés; sin embargo, se mostrado (a lo largo de las diferentes pruebas) que podría establecerse el *Marco* con conexiones asimétricas *ADSL* típicas de uso doméstico, en lo relativo a las necesidades de *ancho de banda* de las sedes (*ERCs*) con respecto a la sede central (*ESCC*).

La configuración establecida mediante *CAR* permite garantizar un *ancho de banda* limitado ya que el tráfico que excede se le puede aplicar una política diferente, en el *proyecto fin de carrera* es descartado y, por tanto, no se envía. Es decir, permite controlar la velocidad máxima del tráfico enviado o recibido en una interfaz.

La configuración se establece a partir del comando *rate-limit* en cada sub interfaz. El uso del comando es el siguiente:

```
rate-limit {input | output} [access-group [rate-limit] acl-index |
qos-group qos-number] <CIR> <Bc> <Be> conform-action <action> exceed-
action <action>
```

Siendo, en la configuración utilizada en el proyecto final de carrera:

input/output: Indica si se aplica al tráfico entrante o saliente al interfaz (o sub interfaz).

acl-index: Número de la lista de acceso que aplica. En el contexto del proyecto definirá el tipo de tráfico al que afecta (en función de la dirección *IPv4*).

<CIR>: Determina la velocidad de transmisión en promedio. El tráfico que está por debajo de esta velocidad cumple con las condiciones ya que es la tasa con la que se van renovando las fichas.

<Bc>: El tamaño de la ráfaga normal en bytes.

<Be>: Cuando el tamaño de la ráfaga supera al de <Bc>, se permite una ráfaga adicional cuyo tamaño es la diferencia entre el <Be> y el <Bc>.

<action>: Son las medidas a tomar en función del tipo de tráfico. Permite una *lista de valores* para aplicar la política deseada (*continue*, *drop*, *set-prec-continue*, *set-prec-transmit* y *transmit*); sin embargo, en el proyecto se utilizarán *transmit* y *drop*.

La configuración del router, se establece con los siguientes criterios recomendados por el fabricante del router para *ISPs*:

```
normal burst <Bc> = CIR * (1 byte)/(8 bits) * 1.5 seconds
extended burst <Be> = 2 * Bc
```

De este modo, se espera que la configuración sea realista ante situaciones en las que haya una WAN en el camino entre la sede y el ESCC de modo que se permitirán ciertas ráfagas en la configuración del protocolo.

Así, para la configuración de cada sede se ha de establecer una serie de pasos en la configuración del router:

1. Establecer la *Lista de Acceso* a la cual se aplicará la restricción de *ancho de banda (CIR)*.

```
access-list 100 permit ip any any
access-list 101 permit ip any any
access-list 102 permit ip any any
```

2. La *Lista de Acceso* se limita en cada sub interfaz y, por tanto, aplicará a cada direccionamiento *IPv4* de la subred. Cuando se aplica la *lista de acceso* se especifican la *política* que limitan el *ancho de banda* utilizado:

```
!ERC-1
interface FastEthernet 0.100
!BW upload (input) de la Sede
  rate-limit input access-group 100 <CIR-UL> <Bc-UL> <Be-UL>
conform-action transmit exceed-action drop
!BW download (output) de la sede
  rate-limit output access-group 100 <CIR-DL> <Bc-DL> <Be-DL>
conform-action transmit exceed-action drop

!ERC-2
interface FastEthernet 0.101
!BW upload (input) de la Sede
  rate-limit input access-group 101 <CIR-UL> <Bc-UL> <Be-UL>
conform-action transmit exceed-action drop
!BW download (output) de la sede
  rate-limit output access-group 101 <CIR-DL> <Bc-DL> <Be-DL>
conform-action transmit exceed-action drop

!ERC-3
interface FastEthernet 0.102
  rate-limit input access-group 102 <CIR-UL> <Bc-UL> <Be-UL>
conform-action transmit exceed-action drop
  rate-limit output access-group 102 <CIR-DL> <Bc-DL> <Be-DL>
conform-action transmit exceed-action drop
```

Por tanto, si el tráfico es menor que el CIR (<CIR-UL/DL>) se trasmite y si queda dentro de la ráfaga normal comprometida (<Bc-UL/DL>) se considerará como tráfico de tipo *conform* y también se transmitirá, mientras que en el caso de que el tráfico supere la ráfaga normal (sin exceder <Be-DL>) se considerará como *Exceed* y se le aplicará otra política diferente, en el caso de la configuración de proyecto se descartará y, por tanto, será considerado como si se hubiese superado (*violate*).

B.III. Configuración de los Routers y el Switch

Se ha establecido una configuración genérica de los distintos dispositivos de red de modo que no se ha tenido que realizar ninguna modificación al pasar de un *Espacio de Trabajo* a otro.

A continuación, se presentan la configuración de los equipos en el esquema que se ha presentado en el apdo. 8.2.3 Espacio de Trabajo Distribuido.

B.III.1. Router A

Como se ha comentado el *Router A* se utiliza de forma única en el primer *bloque* de pruebas en el que se utiliza elementos *hardware* para emular la red. Este router realiza el encaminamiento de las diferentes subredes en ambos los diferentes espacios de trabajo definidos cuando se incorporan elementos hardware de red.

A continuación se presenta la configuración provista al *Router A*, el cual se ha denominado *GW-1721*, describiendo en el nombre el propio modelo del equipo:

```

GW-1721#sh run
Building configuration...
Current configuration : 1801 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname GW-1721
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$v6ib$njn1cnvEXfMGnKQ5GbcuN.
!
no aaa new-model
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
!
!
!
!
ip cef
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
!
!
!
!
interface ATM0
 no ip address
 shutdown
 no atm ilmi-keepalive
 dsl operating-mode auto
!
interface FastEthernet0
 no ip address
 speed auto
!
interface FastEthernet0.1
 encapsulation dot1Q 1 native
 ip address 192.168.1.2 255.255.255.0
!
interface FastEthernet0.100
 encapsulation dot1Q 100
 ip address 192.168.100.1 255.255.255.0
 rate-limit input access-group 100 4192000 786432 1572864 conform-action
transmit exceed-action drop

```



```

        rate-limit output access-group 100 4192000 786432 1572864 conform-
action transmit exceed-action drop
!
interface FastEthernet0.101
 encapsulation dot1q 101
 ip address 192.168.101.1 255.255.255.0
!
interface FastEthernet0.102
 encapsulation dot1q 102
 ip address 192.168.102.1 255.255.255.0
 rate-limit input access-group 102 4192000 786432 1572864 conform-action
transmit exceed-action drop
        rate-limit output access-group 102 4192000 786432 1572864 conform-
action transmit exceed-action drop
!
interface Serial0
 bandwidth inherit 4096
 ip address 10.10.10.1 255.255.255.252
 encapsulation ppp
 clock rate 1000000
!
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 192.168.1.1
ip route 192.168.103.0 255.255.255.0 10.10.10.2
no ip http server
!
!
access-list 100 permit ip any any
access-list 101 permit ip any any
access-list 102 permit ip any any
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
 password 7 11191F06161606050A7B
 login
!
end

```

Tan sólo hay que indicar la interfaz utilizada para establecer la comunicación con la subred VLAN103, la cual emula una sede remota; el resto de subredes son conocidas por el propio equipo debido a que tiene asignada una *subinterfaz* para cada subred considerada (*interface FastEthernet0.1*, *interface FastEthernet0.100*, *interface FastEthernet0.101*, *interface FastEthernet0.102*). Estas *subinterfaces* son el *router por defecto* o *puerta de enlace* (default Gateway) al que envían los paquetes el resto de equipos de la red para establecer el encaminamiento.

Como se puede observar, en la configuración presentada, se establece el encapsulamiento mediante el protocolo 802.1q (*dot1q*) [149] y se ha de considerar que el equipo al que se conecta esta interfaz debe configurarse para establecer este protocolo de comunicación a nivel de enlace de datos.

Se ha configurado una VLAN nativa (*VLAN1*), la cual podría considerarse como *subred de supervisión* si el entorno de pruebas se mantuviese aislado; sin embargo, se utiliza para acceder al router doméstico cuya dirección IP es la 192.168.1.1 el cual proporciona una salida real a *Internet* y que nos ayuda a la hora de actualizar los equipos.

El interfaz *Serial0* es utilizado para emula una *WAN* dentro de una hipotética situación en la que el *Espacio de Trabajo* sea *distribuido* y haya que establecer la comunicación entre los diferentes equipos con una *ancho de banda* limitado. En este caso el equipo marca el ancho de banda ya que sería el *DCE* de la comunicación con el Router B.

B.III.2. Router B

La configuración el *Router B* es la más sencilla de todos los equipos que intervienen en el *entorno de pruebas* del *Marco* implementado. A continuación se muestra su configuración, la cual se utiliza al simular un *Espacio de Trabajo Distribuido* (apdo. 8.2.3 Espacio de Trabajo Distribuido):

```
GW-1750#sh run
Building configuration...

Current configuration : 627 bytes
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname GW-1750
!
enable password 7 044B0D050E2541470748
!
!
!
!
!
memory-size iomem 25
ip subnet-zero
!
!
!
!
interface Serial0
bandwidth 4096
ip address 10.10.10.2 255.255.255.252
encapsulation ppp
!
interface FastEthernet0
ip address 192.168.103.1 255.255.255.0
speed auto
full-duplex
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.10.10.1
no ip http server
!
!
line con 0
password 7 105E0F1A04131F020255
line aux 0
line vty 0 4
password 7 044B0D050E2541470748
login
!
end
```

La configuración de este equipo representa la *puerta de enlace* desde la sede remota para establecer un *Espacio de Trabajo Distribuido*. La interfaz *Fastethernet0* se conecta al Swithc a una boca configurada con la *VLAN103*, la cual se mantiene aislada del resto de *VLANs*

configuradas en el equipo (como veremos en el apartado donde se describe la configuración del *Switch*: apdo. anexo. B.III.3 *Switch*). De este modo, los equipos conectados a la *VLAN103* del *Switch* deberán establecer la comunicación con el resto de *VLANs* mediante este router (*Router B*), el cual ha sido configurado para encaminar todo el tráfico mediante el enlace *Serial0*: de este modo se establece la comunicación a través de la simulación de una *WAN* simulada con el enlace *Serial* a la sede central (*ESCC*).

B.III.3. *Switch*

Se ha realizado la configuración del *Switch* de modo que se pueda utilizar para los todos *Espacios de Trabajo* utilizados. De esta forma se puede pasar de una batería de pruebas a otra sin más que conectar de forma correcta los equipos en cada uno de los puertos del *Swicth*.

La configuración introducida al *Switch* utilizado, denominado *SW-2950*, se presenta a continuación:

```
SW-2950#sh run
Building configuration...

Current configuration : 4436 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname SW-2950
!
enable secret 5 $1$d9ii$ffFXnOyNhda3PNVUcCfoaQ0
enable password 7 095C480A18011A1B055D
!
ip subnet-zero
ip dhcp excluded-address 192.168.100.1 192.168.100.127
!
ip dhcp pool VLAN1
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!
!
interface FastEthernet0/1
switchport access vlan 100
switchport mode access
spanning-tree portfast
!
interface FastEthernet0/2
switchport access vlan 100
switchport mode access
spanning-tree portfast
!
interface FastEthernet0/3
switchport access vlan 100
switchport mode access
spanning-tree portfast
!
```

```
interface FastEthernet0/4
  switchport access vlan 100
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/5
  description VLAN 101
  switchport access vlan 101
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/6
  description VLAN 101
  switchport access vlan 101
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/7
  description VLAN 101
  switchport access vlan 101
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/8
  description VLAN 101
  switchport access vlan 101
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/9
  description VLAN 102
  switchport access vlan 102
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/10
  description VLAN 102
  switchport access vlan 102
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/11
  description VLAN 102
  switchport access vlan 102
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/12
  description VLAN 102
  switchport access vlan 102
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/13
  description VLAN 103
  switchport access vlan 103
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/14
  description VLAN 103
  switchport access vlan 103
  switchport mode access
  spanning-tree portfast
!
interface FastEthernet0/15
  description VLAN 103
  switchport access vlan 103
```

```
        switchport mode access
        spanning-tree portfast
    !
interface FastEthernet0/16
    description VLAN 103
    switchport access vlan 103
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/17
    description Vlan 1
    switchport mode access
    !
interface FastEthernet0/18
    description Vlan 1
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/19
    description Vlan 1
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/20
    description Vlan 1
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/21
    switchport access vlan 2
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/22
    switchport access vlan 2
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/23
    description Puerto SPAN
    switchport access vlan 2
    switchport mode access
    spanning-tree portfast
    !
interface FastEthernet0/24
    description Puerto SPAN
    switchport access vlan 2
    switchport mode access
    !
interface GigabitEthernet0/1
    description CONEXION 1720 dot1q
    switchport trunk allowed vlan 1,100-102
    switchport mode trunk
    !
interface GigabitEthernet0/2
    description CONEXION 1750 access
    switchport access vlan 103
    switchport trunk allowed vlan 1,103
    switchport mode access
    !
interface Vlan1
    ip address 192.168.1.3 255.255.255.0
    no ip route-cache
    !
    !
interface Vlan100
    ip address 192.168.100.2 255.255.255.0
    no ip route-cache
```

```

    shutdown
    !
interface Vlan101
    no ip address
    no ip route-cache
    shutdown
    !
interface Vlan102
    no ip address
    no ip route-cache
    shutdown
    !
interface Vlan103
    no ip address
    no ip route-cache
    shutdown
    !
ip default-gateway 192.168.1.2
ip http server
    !
line con 0
    password 7 071F274F4F0D140C1943
line vty 0 4
    password 7 131511110A0809232575
    login
line vty 5 15
    password 7 131511110A0809232575
    no login
    !
    !
    !
monitor session 1 source interface Fa0/1 - 20
monitor session 1 destination interface Fa0/24
monitor session 2 destination interface Fa0/22
end

```

El aspecto de configuración más destacable es la división de las diferentes ‘*bocas*’ del *Switch* en subredes a través del protocolo *VLAN*. De este modo, mediante el enlace *trunk* hacia el *Router A* (GW-1721) se consigue enviar el tráfico a las diferentes subredes por esta conexión. Para ello se utiliza el protocolo *801.1q* (*dot1q*) [149] [150] en la conexión *GigabitEthernet0/1* al cual debe configurarse el interfaz *FastEthernet0* del *Router A* en el otro extremo.

Otro aspecto que podría ser destacable dentro de la configuración es la configuración de una interfaz, *FastEthernet0/23* como destino de todo el tráfico que circula por el *Switch* (modo *SPAN*); de este modo, con un equipo informático con la tarjeta de red configurada en modo promiscuo se puede realizar un análisis del tráfico de todas las subredes.

B.IV. Configuración de la aplicación VLC como Emisor Multicast

La configuración del equipo emisor mediante la *aplicación VLC* es muy sencilla y tan sólo se han de seguir los pasos que ofrece un asistente. La configuración se realiza sobre el *equipo anfitrión* conectado a la misma LAN que el equipo virtualizado *ERC*. De este modo se configura el *ESVLC* de forma local a la sede.

B.IV.1. Elección de la fuente de contenido multimedia.

Se ha de indicar que la fuente de contenido a emitir puede ser muy variada. Desde una lista de archivos multimedia, contenido desde elementos físicos (*CDs, DVDs,...*), protocolos de red capturados (*RTP, RTSP, MMS, HTTP,...*) o dispositivos de captura (emisión del *Escritorio*, tarjetas capturadoras de vídeo, cámaras *USB...*). En las pruebas realizadas se realizó la emisión mediante una cámara *USB* y también un vídeo de tipo *Flash Player (FLV)* de calidad baja (*LQ*):

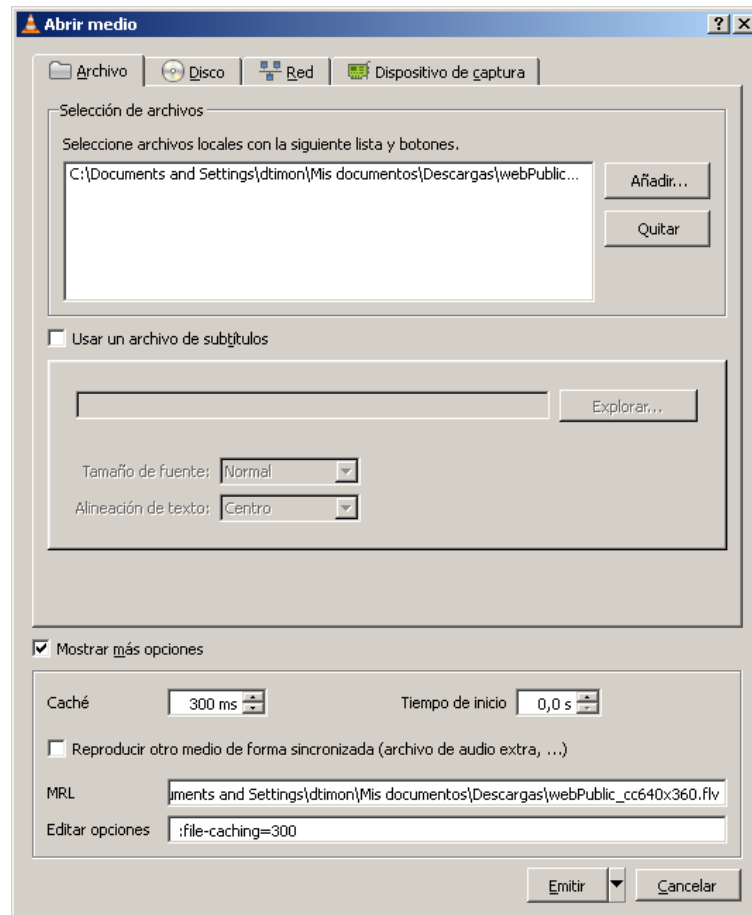


Ilustración 98.-. Pantalla de Selección de la fuente del contenido multimedia en la aplicación VLC. Elección de archivo multimedia local.

Se ofrece la posibilidad de elegir entre una serie de pestañas el tipo de fuente. En la opciones avanzadas (parte inferior) se permite incluir otras opciones de forma manual e incluso emitir un archivo adicional de forma sincronizada en el mismo flujo.

Después de establecer la elección de la fuente del contenido multimedia se ha de seleccionar el tipo de acción que se llevará a cabo. Esta configuración se establece mediante un botón desplegable, el cual tiene varias opciones (*'Encolar', 'Reproducir', 'Emitir' y 'Convertir'*). En el caso de las pruebas se selecciona *'Emitir'*, de modo que al pulsar el botón desplegable nos conducirá a un asistente para configurar la emisión del contenido multimedia.

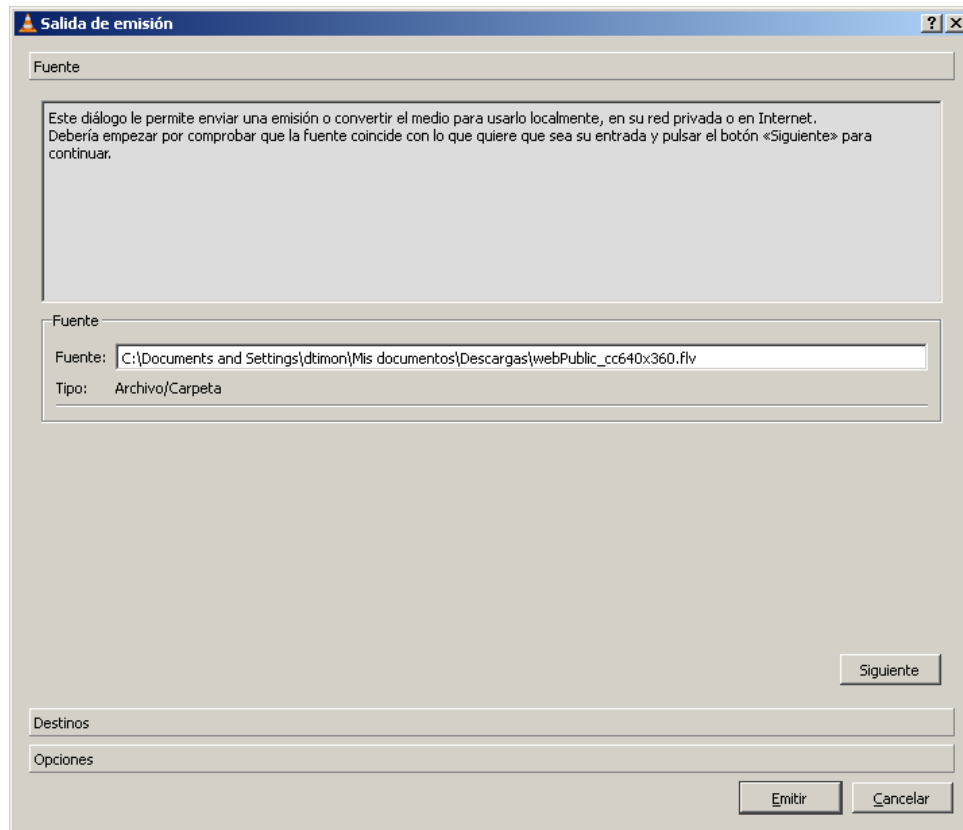


Ilustración 99.-. Asistente de Navegación para la configuración de Emisión de contenido multimedia en la aplicación VLC.

Pulsando el botón '*Siguiente*' nos permite navegar a la pantalla de elección del *destino de la emisión*.

B.IV.2.Destino de la Emisión

En esta pantalla del asistente se ha de seleccionar la forma en la que se establecerá la emisión. Dependiendo de reproductor que la mostrará se tendrá que tener en cuenta ciertos parámetros de acuerdo con las especificaciones del receptor.

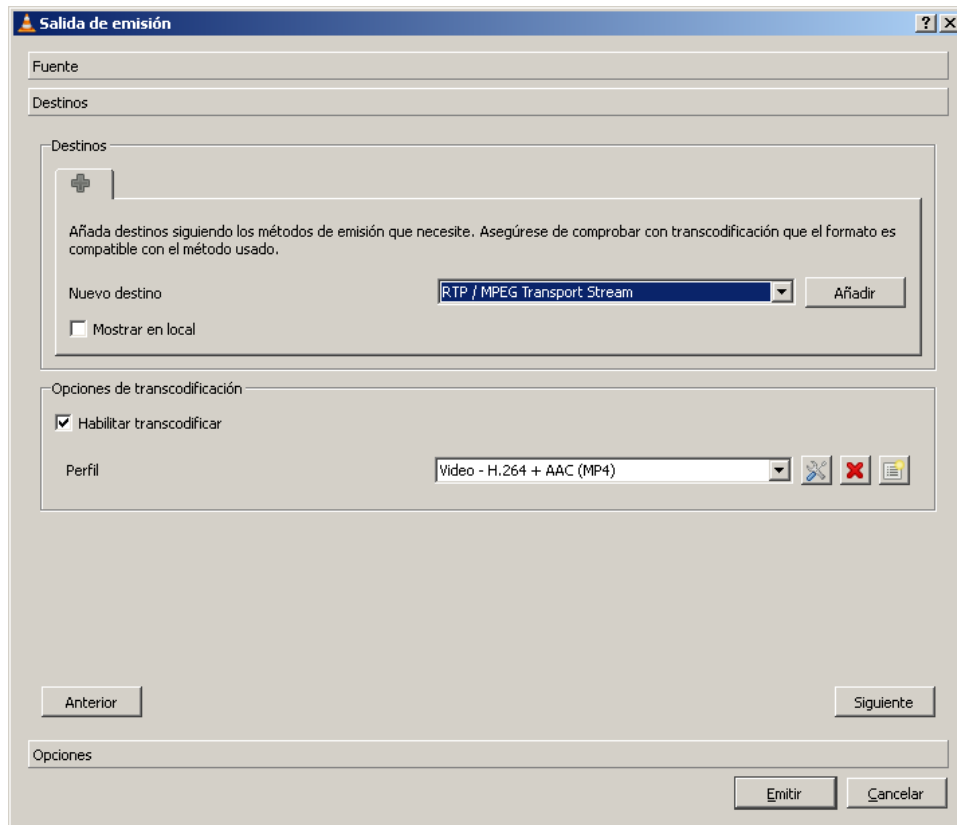


Ilustración 100.-. Selección del Tipo de Destino a Establecer.

Se establece una emisión *RTP/MPEG TS* con el objetivo de establecer un flujo *Streaming Media Multicast*.

A continuación (Ilustración 101), se muestra la configuración establecida en la prueba:

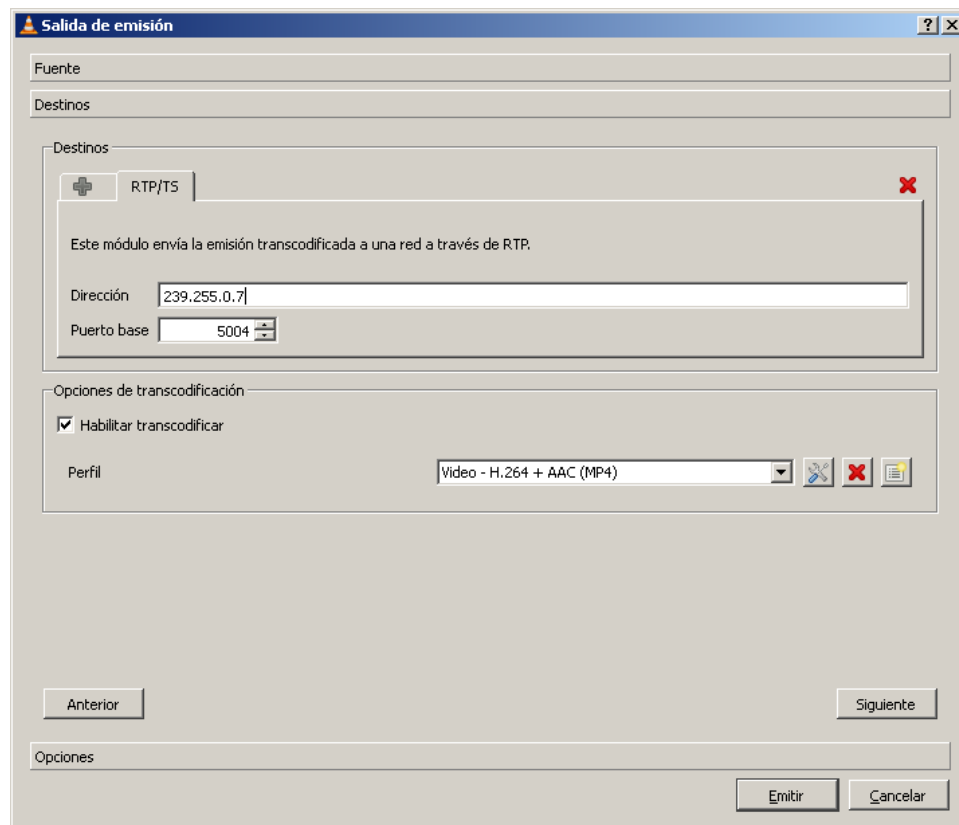


Ilustración 101.-. Configuración del tipo de destino para la emisión del contenido Multimedia mediante tráfico Streaming Media.

Se mantiene el perfil con los valores de la codificación de vídeo y audio por defecto, debido a que, al realizarse la recepción con el propio plugin de *VideoLAN*, se espera que el equipo terminal pueda establecer la decodificación de forma apropiada.

B.IV.3. Establecer las Opciones

Es posible en la configuración establecer otras opciones en la emisión de contenido [52]. En la última pantalla del asistente se ofrecen opciones para la emisión del contenido. En las pruebas establecidas se mantienen sin realizar modificaciones sobre las que proporciona el asistente:

Salida de emisión

Fuente

Destinos

Opciones

Opciones varias

☒ Emitir todas las emisiones elementales

☐ Anuncio de SAP Nombre de grupo

Tiempo de vida (TTL):

Cadena de salida de emisión generada

```
:sout=#transcode{vcodec=h264,vb=0,scale=0,acodec=mp4a,ab=128,channels=2,samplerate=44100}:duplicate{dst=rtp{dst=239.255.0.7,port=5004,mux=ts},dst=display}:no-sout-rtp-sap:no-sout-standard-sap:sout-all:t看=1:sout-keep
```

Anterior

Emitir Cancelar

Ilustración 102.-. Opciones finales de la emisión.

Al pulsar el botón emitir, la aplicación comienza la emisión según la configuración.

C Manual del Usuario Administrador

El siguiente manual es una guía donde se describen los pasos necesarios que se deben dar para realizar una correcta configuración del sistema de *Difusión Selectiva* desarrollado.

Se va a plantear la configuración del sistema como la configuración de dos tipos de equipos: *ESCC* (*Equipo Servidor de Configuración y Contenido*) y *ERC* (*Equipo Reproductor Cliente*).

Antes de comenzar con la configuración del sistema implementado, se recomienda tener ciertas nociones del sistema y conocer ciertas definiciones establecidas en el *proyecto final de carrera*. Esto es importante debido a que se utilizará cierta nomenclatura definida en el *proyecto fin de carrera* y que puede llevar a confusión a un administrador que no esté familiarizado con éste.

A continuación, se muestra cómo se debería realizar la configuración del *ESCC*:

C.I. Manual del Usuario Administrador del ESCC

El primer paso que hay que dar para acceder a la configuración del *ESCC* es escribir en el navegador la *dirección IPv4*⁴² del *servidor* de modo que el propio equipo redirigirá el tráfico hacia la misma dirección utilizando el protocolo *https*. Por tanto, nos pedirá la aceptación de un certificado que garantiza que la conexión se establece de forma segura.

Posteriormente y una vez aceptado certificado se mostrará la pantalla de acceso a la configuración.

⁴² Si fuera la primera vez que se accede al equipo o no se recordara la dirección IPv4 que se configuró, se ha establecido la posibilidad de acceder a éste a través de un cable directamente conectado a la interfaz de red a la dirección 169.254.69.13.



Ilustración 103.-. Pantalla de acceso de configuración del sistema.

Dependiendo del usuario con el que se acceda al sistema aparece un menú u otro, siendo el usuario 'sistema' el usuario que puede acceder a todas las páginas y a todo el menú:

- **admin**

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-------------	----------	-------

- **sistema**

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

El contenido de los menús se mantiene igual para cada uno de los usuarios que accede al sistema. En los siguientes sub apartados se detalla cada una de las opciones de menú definidas.

C.I.1. MENÚ UBICACIONES

El primer menú que se debe configurar en los relativo a la aplicación de difusión selectiva es el menú 'UBICACIONES'. A continuación se muestra el menú completo.



Ilustración 104.-. Menú Ubicaciones en el ESCC

El orden de configuración del sistema debe ser descendente, es decir:

- Creamos y/o seleccionamos la *Ubicación* que se va a configurar.
- Configuramos la *Ubicación*:
 - Definimos las *Zonas* en los distintos *Escenarios*.
 - Añadimos *Listas de Reproducción* a cada *Zona* del *Escenario* seleccionado.
 - Asignamos los *Archivos Multimedia* a cada *Lista de Reproducción* de cada *Zona* de cada *Escenario*.
- Asignamos cada *ERC* a una determinada *Ubicación*.

Seguidamente se presentan cada uno de los elementos del menú:

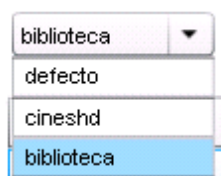
UBICACIONES >> Seleccionar/Crear Ubicación



Ilustración 105.-.Pantalla de Presentación donde se selecciona/crea la Ubicación a configurar

En la ilustración anterior se pueden identificar diferentes regiones, las cuales se han etiquetado mediante texto según sean comunes a otros menús ('Info. Ubicación' e 'Info Socket'), o de forma numérica: 1, 2, 3 y 4, si es específico a esta pantalla de configuración.

- **Región Info. Ubicación:** En esta región se muestra qué *Ubicación* se está configurando. Como veremos posteriormente, esta etiqueta aparecerá en todas las páginas donde la *Ubicación* sea determinante en la configuración que se ha seleccionado. Inicialmente la primera vez que se accede aparece la ubicación 'defecto' como se observa en la Ilustración 105. De este modo, si no se especifica ningún valor, se configura la *Ubicación* denominada 'defecto', la cual está siempre presente en el sistema y no puede ser eliminada.
- **Región 1:** Con este desplegable ('*ComboBox*') se selecciona la *Ubicación* que vamos a configurar en el resto de páginas de configuración. Al pinchar sobre él se despliegan todas las *Ubicaciones* que han sido creadas en el sistema con anterioridad (ver Región 3 para el mantenimiento de *Ubicaciones*):



Una vez se selecciona un *Ubicación* en la región 1; la región 'Info. Ubicación' se actualiza de forma automática:

Ubicaciones

ubicación: biblioteca

- **Región 2:** Esta región se ha subdividido en dos subregiones: 2a y 2b. Éstas muestran información de la *Ubicación* y según se seleccione en el listado de la región 2a se muestra información de los *ERCs* suscritos a esta *Ubicación* en la región 2b. No obstante, esta región no modifica la *Ubicación* que se está configurando simplemente presenta información.

En la subregión 2b se muestra la *dirección IPv4* de los *ERCs* suscritos:

ID	Nombre	N Equipos Asociados
1	defecto	0
33	cineshd	0
39	biblioteca	1

Región 2a

ID	IP	Mascara
30	192.168.1.8	255.255.255.255

Región 2b

- **Región 3:** Permite crear nuevas ubicaciones y eliminar ubicaciones anteriores. Cuando se elimina una ubicación, los equipos suscritos a dicha ubicación pasan a formar parte de la *Ubicación* 'defecto'. Antes de crear una nueva *Ubicación* se informa al administrador del sistema que esta operación se va a llevar a cabo, pidiendo confirmación para la acción:

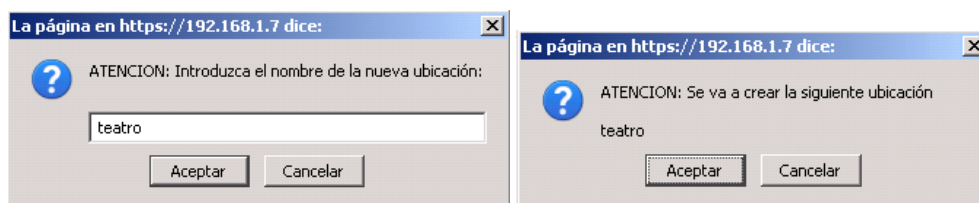


Ilustración 106.-. Pantallas emergentes en la creación de nuevas Ubicaciones en el sistema.

Cuando se selecciona una *Ubicación* en la región 2a y posteriormente se pulsa sobre el botón eliminar, como parte del mantenimiento, se pide la confirmación para llevar a cabo esta acción. Además se informa que los equipos asociados a la *Ubicación* que se va a eliminar van a ser asignados a la *Ubicación* 'defecto':

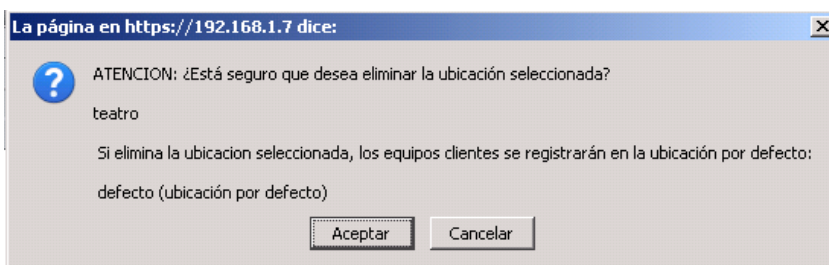
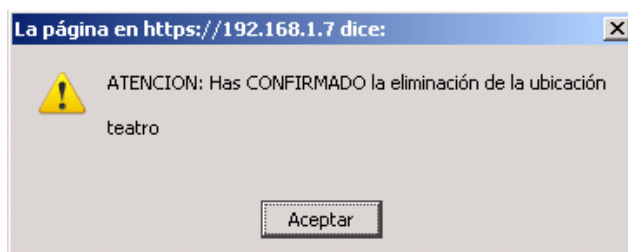
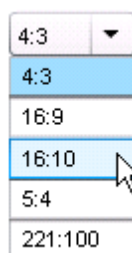


Ilustración 107.-. Pantalla emergente en la eliminación de Ubicaciones en el sistema.

Finalmente cuando se elimina una *Ubicación* se muestra un nuevo mensaje informativo:



- **Región 4:** Permite realizar una previsualización de una determinada *Ubicación*. La previsualización se activa al cambiar la relación de aspecto o pulsando sobre la representación del SPC (imagen de monitor). Dependiendo del tipo de pantalla a la que esté conectada la salida del vídeo de nuestro equipo (ERC) se puede realizar una previsualización en diferentes formatos:



Al cambiar de formato se abrirá una ventana en el navegador en el formato escogido, mostrando la previsualización de la *Ubicación* que esté seleccionada en la región 2a:



Ilustración 108.-. Pantalla Emergente de Previsualización en formato 4:3.

Si se pulsa sobre la imagen del *SPC* representado en la *Interfaz Gráfica de Usuario* se abrirá también la pantalla emergente de previsualización.

- **Región Info. Socket:** Muestra información del estado del *socket*. Este panel se muestra dentro de varias pantallas de configuración. Sirve para conocer si se están enviando órdenes a los *ERCs* o el *socket* se mantiene activo. En caso de pérdida de conexión con el *socket* aparecerán informes en esta región de intentos de reconexión constantes cada *30 segundos*.

Info Socket:

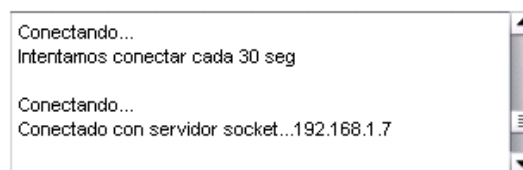


Ilustración 109.-. Se ha perdido la conexión con el servidor socket. Al pasar 30 segundos se vuelve a intentar obteniendo el acceso.

Adicionalmente, cuando se elimina una *Ubicación* sobre esta pantalla, se envía una orden al *socket*. Esta orden indica a los *ERCs* que estaban conectados a dicha *Ubicación* reenvían una petición de configuración al servidor y así obtener la configuración de la *Ubicación* 'defecto'.

Info Socket:

Datos leídos del Servidor:

```
<actualizar ubicacion='teatro'>teatro</actualizar>
```

Ilustración 110.-. Orden enviada desde el ESCC. Los ERCs asociados a la *Ubicación* 'teatro' deben pedir de nuevo la configuración del *Ubicación*.

Esta orden es importante para el correcto funcionamiento de un *ERC* asignado a una *Ubicación* que desaparece de la configuración del sistema.

UBICACIONES >> Configurar Ubicación >> Zonas a Esc

Una vez que se ha creado una *Ubicación* se ha de pasar a definir las *Zonas* dentro de un determinado *Escenario*. El sistema implementado permite tener pre-configurado hasta cuatro *Escenarios* por *Ubicación*.

A continuación, se presenta la pantalla de configuración del *Escenario* de una *Ubicación*. En esta pantalla es donde se definen las *Zonas* del *Escenario* concreto de la *Ubicación* a configurar:

Ilustración 111.-. Pantalla de Definición de las Zonas asignadas a cada uno de los posibles Escenarios de una *Ubicación*.

La pantalla se divide en diferentes regiones, las cuales se describe a continuación:

- **Región Info. Ubicación:** Muestra la *Ubicación* que estamos configurando y que afecta de forma directa a la configuración de esta página.
- **Región 1:** Muestra el *Escenario* que se está configurando y la *Zona* que se está dimensionando. Es posible mantener pre-configurado hasta cuatro *Escenarios* (región 3b) y cada *Escenario* puede ser dividido en cuatro *Zonas*. Estos desplegables ('ComboBox') establecen qué se está configurando (*Escenario*, *Zona*); sin embargo, para indicar que un determinado *Escenario* se aplica a la *Ubicación* se establece con los 'radioBotton' de la región 3b.
- **Región 2:** Se ha subdividido en tres subregiones ya que esta región ofrecen la posibilidad de modificar la configuración y visualizar de forma aproximada como resultaría el *Escenario*. La región 2a permite configurar el tamaño y la posición de la *Zona* que se está configurando. Los valores aquí introducidos se deben referenciar en % y son con respecto al tamaño final del *Escenario*. La región 2b nos proporciona la configuración que tendrá el fondo, la cual es posible aplicar imágenes o simplemente mostrar un color plano.
- **Región 3:** Es la región donde se aplican los cambios realizados sobre cada *Escenario*. Se subdivide en tres regiones:
 - La región 3a permite guardar la configuración realizada sobre el *Escenario* y la *Zona* seleccionada en la región 1. Permite, mediante el botón 'Resetear', borrar todas las *Zonas* y aplicar un fondo negro a la configuración.
 - La región 3b aplica el *Escenario* a la *Ubicación* específica.
 - El marcado ('checkBox') de la región 3c permite que se guarden los cambios en la *base de datos* cada vez que se realiza alguna modificación. El flujo de información se hace mayor ya que cada cambio que se realice se graba en la configuración del servidor.
- **Región Info:** Muestra la prioridad de cada una de las capas de modo que según sea la capa superpondrá sobre la otra.

NOTA: Si se requiere un fondo dinámico a la configuración de una determinada Ubicación, es posible obtenerlo estableciendo la Zona-D a tamaño completo en el Escenario y programar aquí la lista de archivos multimedia que se quiere que se reproduzcan. Así el fondo se establece como dinámico en la presentación.

Aparecen otros botones sombreados y marcados con elipses azules discontinuas los cuales se aplicarán en futuras versiones del software.

UBICACIONES >> Configurar Ubicación >> Listas a Zonas

En esta pantalla se asocian las *listas de reproducción* de los archivos a cada *Zona*. Las *listas de reproducción* son exclusivas de cada *Ubicación*.

Se ha de configurar una fecha en la cual se empezará a reproducir una determinada *lista de reproducción* en una determinada *Zona*. Dentro de la *lista de reproducción* se definirán los intervalos horarios donde se reproduce cada archivo.

zonasListas

Inicio UBICACIONES > ARCHIVOS > ACTUALIZAR > SISTEMA > DESCARGAS > MANUAL > Salir

1 ESC-0 ZONA-A

2

ID	Lista	Fecha de Inicio
105	uc3m_teatro	2009-08-09

3

Listas Disponibles	Núm. Archiv
defecto	1
uc3m_teatro	5

4

Eliminar Lista Nueva Lista

5a

Info. Ubicación ubicación: biblioteca

ID	Archivos en la Lista Selec	Hora Inicio	Hora Fin
1	uc3mVWell2	00:00:00	23:59:00
2	uc3mVWell1	00:00:00	23:59:00
3	uc3mVS1	00:00:00	23:59:00
4	uc3mVS2	00:00:00	23:59:00
5	uc3mVS3	00:00:00	23:59:00

5b

Día: 9 Mes: Noviembre Año: 2009

2009-08-09 LUNES

Algoritmo: Secuencial

Transición: Sin-Transición

5c

Nombre Lista: uc3m_teatro

Editar Quitar

Ilustración 112.-. Pantalla de asociación de *Listas de Reproducción* a *Zonas* específicas de *Escenarios* en cada ubicación.

- **Región Info. Ubicación:** Esta región muestra las *Ubicación* que se está configurando al igual que en el resto de pantallas.
- **Región 1:** En esta sección nos encontramos dos elementos desplegables ('comboBox'). Sin embargo, se ha decidido desactivar la capacidad de cambiar el *Escenario* de modo que sólo aparece el que se ha configurado para la *Ubicación* para así no llevar a confusión al administrador del sistema. Además, se ha de tener en cuenta que las listas que se asocian a cada *Zona* son independientes a cada *Escenario*; es decir, la *Zona A* del *Esc-0* es independiente a la *Zona-A* del *Esc-1*.

ESC-1 ZONA-A

ESC-1 ZONA-A

B-NO EXISTE

C-NO EXISTE

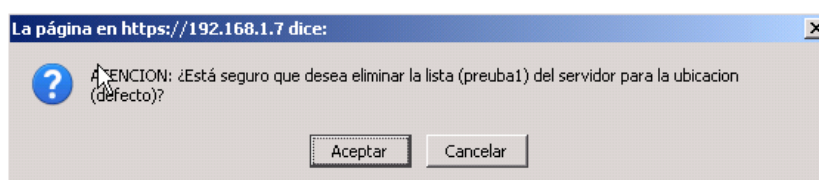
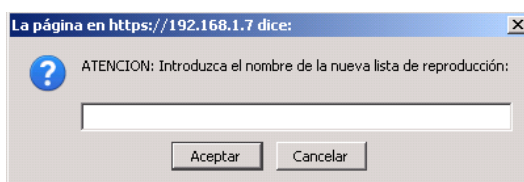
D-NO EXISTE

La aplicación también tiene la capacidad de informar en el desplegable de que la *Zona* no se ha definido. Así, el administrador del sistema sabe en cada momento que una determinada *lista de reproducción* se está aplicando a una determinada *Zona* y si tiene la posibilidad de aplicarla a otra zona distinta.

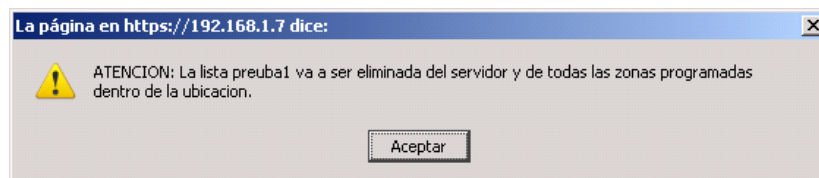
- **Región 2:** Esta región muestra las *listas de reproducción* asociadas a cada *Zona*. Según se pulse sobre una de las listas de esta *Zona* se mostrará información y se activarán una serie de campos sobre la región 5c (en este caso se activan las

opciones 'Editar' y 'Quitar'). El algoritmo y la transición se aplica a la configuración de cada *Zona* y no a la lista de reproducción en sí; es decir, si se programara el algoritmo⁴³ de reproducción estos parámetros, los cuales se almacenan en la relación (*Zona, lista de reproducción, Ubicación*) aplican sólo a la configuración de esa *Zona*.

- **Región 3:** Muestra todas las *listas de reproducción* configuradas para esta *Ubicación*. Según se pulse sobre una de las listas se mostrará información en la región 5a acerca de los archivos asignados del mismo modo que al seleccionarla en la región 2. Al seleccionar una lista en esta región se activarán la funcionalidad de los botones sobre la región 5c (activación 'Añadir' sobre 'Editar').
- **Región 4:** Los botones de esta región permite eliminar *listas de reproducción* creadas para una determinada *Ubicación* o crear otras nuevas disponibles en ella. Será, por tanto, donde se realice el mantenimiento de las lista de reproducción.



Cuando se elimina una lista asociada a alguna *Ubicación* esta se elimina también de todas las *Zonas* a las que está asociada. Se informa al administrador del sistema que se va a realizar la operación:



- **Región 5:** Esta región muestra información acerca de la *lista de reproducción*. Además muestra diferentes botones según se está editando una lista ya programada sobre una *Zona* (región 2) o si se ha seleccionado una lista de forma general dentro de la *Ubicación* (región 3) y se quiere aplicar a una de las *Zonas*.
Así, cuando seleccionamos una *lista de reproducción* de forma general, el botón que se activan en la región 5c es 'Añadir':

⁴³ En principio esto parámetros se mantienen desactivados y deberán ser definidos en evoluciones futuras; no obstante, si se establece la relación y se configura en la relación: *Ubicación-Lista-Zona(Escenario)*.

Nombre Lista: uc3m_promo

Editar

Añadir

Mientras que si pulsamos sobre una lista asociada a una *Zona* son los que se muestran en la Ilustración 112: 'Editar' y 'Quitar'.

Existen algunos botones que posiblemente sean activados en posteriores versiones del software como (como se ha comentado); éstos son el algoritmo de reproducción de la lista y la transición entre elementos. Estos botones quedan en principio desactivados.

UBICACIONES >> Configurar Ubicación >> Archivos a Listas

Una vez se han definido las listas y se han asociado a las *Zonas* del *Escenario* configurado para la *Ubicación*, falta por añadir los archivos multimedia que se quieren reproducir a las listas programadas. Posteriormente en el Menú ARCHIVOS (Apéndice C.I.2) se explicará como subir archivos al servidor, por lo que en este apartado supondremos que se dispone de archivos ya almacenados en el *ESCC* para poder asociarlos a las listas.

A continuación se muestra la pantalla en la que se definen los intervalos de reproducción de los diferentes archivos en cada *lista de reproducción* de una determinada *Ubicación*:

Inicio UBICACIONES > ARCHIVOS > ACTUALIZAR > SISTEMA > DESCARGAS > MANUAL > Salir

pfc v1.0

ListasArchivos

1

anuncios

2

ID	Archivo	Hora Inicio	Hora Fin
1	anuncio1	00:00:00	23:59:00
3	anuncio2toyota	00:00:00	23:59:00
4	anuncioSprite	00:00:00	23:59:00
6	anuncioSprite3	00:00:00	23:59:00

3

TODOS

ID	Archivo	Tipo	Extensión	Ubica
151	Earth_HD	VIDEO	flv	loc
106	uc3mVideoPromo	VIDEO	flv	loc
106	uc3mTecno2	VIDEO	flv	loc
111	uc3mSecondLife	VIDEO	flv	loc
141	Tron_HD	VIDEO	flv	loc
143	The Time Travellers 1	VIDEO	flv	loc
145	MoonTrailer_HD	VIDEO	flv	loc
147	Inglourious Basterds	VIDEO	flv	loc

Info. Ubicación

ubicación: defecto

4a

▶

⏸

◀

🔊 On

4b

Inicio: 0 0 Fin: 0 0

Prioridad: ALTA Tipo: VIDEO

Opcion 1: Normal Opcion 2: 5

Nombre Archivo: Earth_HD

Descripción:

Añade una descripción a la configuración de este fichero... debe rellenar todos los campos para poder añadir un nuevo fichero a la lista...

4c

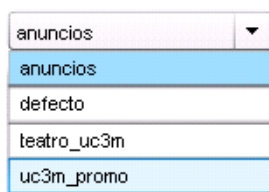
Editar

Añadir

Ilustración 113.- Pantalla de asociación de archivos a listas de reproducción específicas de cada Ubicación.

Clasificamos la página en 5 regiones para así realizar una explicación de forma ordenada:

- **Región Info. Ubicación:** Muestra la *Ubicación* en la que nos encontramos configurando la *lista de reproducción*. Según sea la elección, aparecerá en el desplegable de la región 1 solamente las *listas de reproducción* definidas para esta *Ubicación* (la *Ubicación* se selecciona desde la pantalla presentada en el apartado UBICACIONES >> Seleccionar/Crear Ubicación).
- **Región 1:** El desplegable almacena todas las listas disponibles para la *Ubicación*. Es posible que existan varias *listas de reproducción* con el mismo nombre asociadas a distintas *Ubicaciones*; sin embargo, para una misma *Ubicación* no pueden aparecer dos listas diferentes con el mismo nombre. Tan sólo se podrá acceder a la configuración de la *lista de reproducción* correspondiente a la *Ubicación*. Si se quisiera editar la *lista de reproducción* de otra *Ubicación* antes deberíamos seleccionar la *Ubicación* a configurar (ver apartado UBICACIONES >> Seleccionar/Crear Ubicación). A continuación se muestra qué listas se tienen disponibles en el ejemplo de configuración para la *Ubicación* 'defecto':



región 1

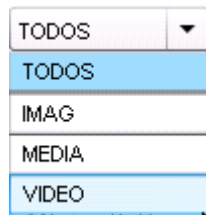
- **Región 2:** Esta región dentro de la página de configuración de las lista muestra los *archivos multimedia* que se han programado para una *lista de reproducción* de una determinada *Ubicación*. Se muestra también el intervalo de reproducción programado dentro de la lista, el cual se define con la hora de inicio y la hora de fin de la reproducción. En el ejemplo, se han definido un intervalo único (00:00:00-23:59:00); sin embargo, es posible definir tantos intervalos como se quiera. El sistema calculará que dicho intervalos no se solapen. Si existen intervalos en los que no se ha programado ningún archivo, el sistema reproducirá los *archivos multimedia* configurados en el intervalo anterior.
- **Región 3:** Esta región lista todos los *archivos multimedia* disponibles en el ESCC. Cuando se selecciona uno de estos archivos en esta región se producen ciertos eventos que provocan que la región 5 varíe. Fundamentalmente se ofrece una previsualización del archivo y también se activan ciertos campos para añadir el archivo seleccionado a la *lista de reproducción* seleccionada.





El listado muestra determinada información acerca del archivo:

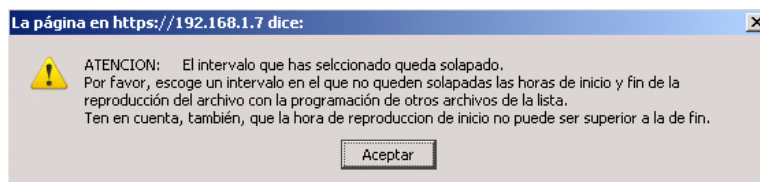
- **Tipo:** tipo de archivo multimedia; puede ser Vídeo o Imagen (en futuras versiones se espera introducir la reproducción de archivos de media⁴⁴: swf).
- **Extensión:** extensión del archivo multimedia.
- **Ubica:** ubicación física del archivo. El valor por defecto es 'loc' e indica que el archivo se encuentra almacenado en el ESCC.

⁴⁴ Se han realizado diferentes pruebas de reproducción de archivos swf; sin embargo, en el periodo de tiempo en el que se programó la aplicación cuando un archivo se había compilado en versiones anteriores a *Flash Professional CS3* la visualización del archivo no correspondía con el programado en el *Escenario*, por lo que se descartó incluir esta posibilidad. Cuando el archivo se había compilado en *Adobe Flash CS3* para *Flash Player 9* (versión en la que se ha programado la aplicación) la visualización era correcta.

En esta región se ha establecido un pequeño filtro para realizar la búsqueda de archivos de forma más cómoda, así se puede indicar que se muestren sólo los diferentes tipos de archivos considerados en el sistema:



- **Región 4:** Esta región es muy importante para proveer la configuración del *archivo multimedia* a la *lista de reproducción*. Se ha subdividido en tres regiones según la funcionalidad establecida:
 - En la región 4a se ha programado un reproductor que nos permite pre-visualizar los archivos multimedia que se quieren introducir en la lista de reproducción. Cuando es una imagen simplemente se muestra; sin embargo, cuando se selecciona un vídeo, se ha dotado al reproductor de los siguientes botones:
 -  **Reproducir:** permite ejecutar la reproducción del vídeo.
 -  **Pausar:** permite pausar la reproducción del vídeo. Si se pulsa el botón de reproducción, vuelve a tomar el tiempo desde el punto donde se pausó.
 -  **Parar:** para completamente la reproducción del vídeo y toma el tiempo de reproducción al inicio del vídeo. Si se vuelve a pulsar el botón reproducción, el vídeo se muestra desde el inicio y no desde la última vez que se pulsó el botón de parada.
 -  **Sonido:** Si se pulsa cambia de estado entre on/off. Si el estado es off se deshabilita el sonido en la reproducción del vídeo, pero sólo en la visualización presente en esta pantalla. Este botón no tiene efecto cuando se reproduce el vídeo en ejecución sobre una *lista de reproducción*.
 - La región 4b es donde se configuran los parámetros relativos al archivo multimedia dentro de la *lista de reproducción*. Estos parámetros se han definido en la estructura de la *base de datos* y se ha implementado su edición. Según el valor de estos parámetros, se reproducirá la lista conforme al algoritmo. Estos parámetros son los siguientes:
 - **Intervalo de reproducción (Inicio-Fin):** El sistema realiza una comprobación del intervalo que se pretende introducir para que así no se produzca solapamiento. El sistema informa ante casuística posible con respecto al solapamiento:



- **Prioridad:** se aprovisiona la capacidad de establecer prioridad sobre el archivo en una *lista de reproducción*. El algoritmo establecido para establecer el orden de reproducción de la lista es el secuencial y reproducirá los archivos atendiendo al orden indicado en la prioridad.

- **Tipo:** Muestra el tipo de vídeo seleccionado.
- **Opción 1 (deshabilitado):** Permite establecer diferentes opciones respecto al archivo. Por ejemplo, podrían programarse diferentes transiciones entre archivos. La funcionalidad se deja para líneas de trabajo futuro.
- **Opción 2:** Se utiliza cuando el archivo seleccionado es una imagen y representa el número de segundos que se muestra en la Zona.

UBICACIONES >> Equipos a Ubicaciones

A continuación se presenta la pantalla donde se configuran los equipos a una determinada *Ubicación*. Esta configuración es la que permite establecer *Difusión Selectiva* y está determinada por la dirección IPv4 del ERC.

Inicio UBICACIONES > ARCHIVOS > ACTUALIZAR > SISTEMA > DESCARGAS > MANUAL > Salir

pfc v1.0 Equipos

1 Configura Ubicacion: biblioteca

2

ID	IP	Mascara
30	192.168.1.8	255.255.255.255

3a

IP:

192 168 1 8

Equipo /32

255 255 255 255

3b

Eliminar Equipo Editar

Cambiar Equipo a UBICACION AC

3c

Insertar

Info Socket

Info. Socket

Se carga http://192.168.1.7:8080/crossdomain.xml
Conectando...
Conectado con servidor socket...192.168.1.7
Datos leídos del Servidor:

Ilustración 114.-. Pantalla de asociación de los ERCs a cada Ubicación.

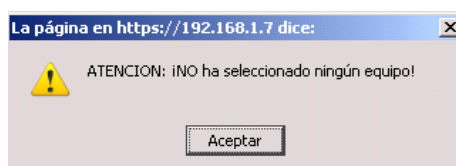
Se divide esta pantalla también en diferentes regiones para así centrar la explicación sobre cada una de ellas:

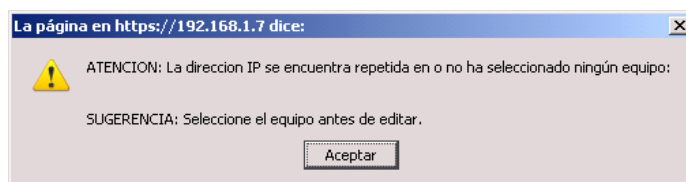
- **Región 1:** Seleccionamos mediante el desplegable (objeto 'comboBox') de la región la *Ubicación* que queremos configurar. En la *Ubicación* podremos asignar nuevos ERCs, eliminarlos, modificarlos o asignarlos a otras ubicaciones a través de la configuración del resto de regiones.
- **Región 2:** Esta región nos permite elegir el ERC configurado dentro de la *Ubicación* para realizar una operación sobre éste. Es posible que este equipo ya no exista debido a que haya cambiado su dirección IPv4 en su configuración local

o se quiera mover a otra *Ubicación*. Será en esta región donde seleccionamos el *ERC* para su tratamiento en la región 3b.

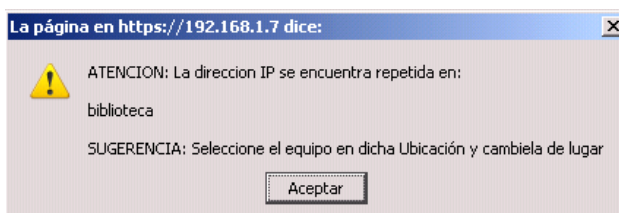
- **Región 3:** Esta región es donde aplicamos la configuración. Por una parte, cuando aplicamos las diferentes órdenes se debe: por una parte, actualizar la *base de datos* y, por otra, indicar a los *ERC* implicado que realicen una determinada acción.
 - La región 3a permite introducir manualmente la *dirección IPv4* del *ERC* que se quiere añadir o editar una ya existente. En esta región siempre se considera que se introducirán equipos uno a uno; no obstante, se ha pensado que en versiones posteriores del sistema se requiera asignar rangos de direcciones *IPv4* a una determinada *Ubicación*. Se ha provisto por tanto el mecanismo para editar la *Zona* de aplicación de la máscara pero sin hacer uso de ella en las peticiones de la configuración de los *ERCs*.
 - En la región 3b se tienen tres opciones donde previamente se ha haber seleccionado un *ERC* en la región 2. Se pueden realizar ciertas acciones:
 - Editar la *dirección IPv4* de un *ERC*, se asignará el valor configurado en la región 3a.
 - Eliminar un *ERC* de una *Ubicación* o cambiarlo de *Ubicación*. El evento de configuración se realiza al pulsar cada uno de los botones:
 - ‘Eliminar Equipo’: al pulsar este botón el sistema elimina el equipo de la *Ubicación* y envía una orden de actualización por si el equipo existiera y así se sincronice con la *Ubicación* ‘defecto’ (se puede visualizar el envío de esta orden en la región Info. Socket).
 - ‘Cambiar Equipo’ a: al pulsar sobre el botón cambiará el equipo a la *Ubicación* previamente seleccionada en el desplegable que le sigue. El sistema envía una orden al equipo para que actualice la configuración a través de una petición al *ESCC* y obtenga, así, la configuración de la nueva *Ubicación* (se puede visualizar el envío de esta orden en la región Info. Socket).

Esta pantalla mostrará mensajes de advertencia con ánimo de advertir al administrador cuando no se haya seleccionado ningún equipo *ERC*.





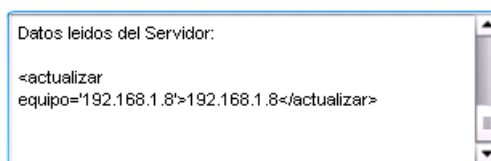
- La región 3c nos permite introducir un nuevo *ERC* en la configuración de la *Ubicación* con tan sólo introducir la nueva dirección *IPv4* sobre la región 3a y pulsar el botón 'Insertar' de la región 3c. El sistema revisa la casuística para que no se repitan configuraciones en diferentes ubicaciones:



- **Región Info. Socket:** La información de las órdenes transmitidas se informan en esta región. Es reseñable que si cambiamos a un equipo de *Ubicación* se debe informar al *ERC* de que debe realizar una nueva petición al *ESCC* para recuperar la nueva configuración.

En versiones iniciales surgió un problema y era que se informaba al equipo mientras se actualizaba la configuración del *ESCC* que producía que en determinadas ocasiones el *ERC* pidiera la nueva configuración antes de que estuviera almacenada la nueva *Ubicación* del *ERC* en la *base de datos* del *ESCC*. Actualmente este importante 'bug' ha sido resuelto de modo que no se escribe en el *socket* hasta que no se ha recibido la confirmación de actualización de la *base de datos*.

Info Socket:



C.I.2. MENÚ ARCHIVOS

La gestión de los archivos que se quieren programar y reproducir se controla desde una interfaz web. Esta característica es importante para así proporcionar al administrador las herramientas necesarias para poder planificar y realizar la programación de contenido multimedia.

Dentro del MENÚ ARCHIVOS se tienen dos funcionalidades importantes, por una parte, en la '*Gestión de Archivos*' se permite la carga, previsualización y eliminación de contenido multimedia y, por otra parte, en la '*Búsqueda de Archivos*' se plantea una búsqueda de los archivos basada en la *Ubicación* y *Lista de Reproducción* de modo que se puede tener un control de forma rápida de la programación de los archivos multimedia en el sistema.



ARCHIVOS >> Gestión de Archivos

La gestión de los archivos es una parte fundamental dentro del *Marco* implementado debido a que facilita subir y eliminar archivos de la programación. Por tanto, permitirá añadir el contenido que será publicado hacia el público objetivo. Es por ello, que se implementa un sistema que proporciona la carga/eliminación de los archivos de forma remota.

A continuación se muestra la pantalla que permite la '*Gestión de Archivos*' dentro del sistema. Los *ERCs* copiarán de forma local los archivos que se suban al servidor mediante esta pantalla, es consecuencia de aplicar la arquitectura *Auto-Proxy*.



Ilustración 115.-. Pantalla de Gestión de los Archivos en el ESCC.

- **Región 1:** Esta región define un pequeño filtro para mostrar los archivos que se han cargado en el *ESCC*. Dependiendo del valor seleccionado en el desplegable se presentará el tipo de archivo. Las posibilidades⁴⁵ se muestran en la siguiente figura:

⁴⁵ La posibilidad de subir archivos MEDIA (swf) se ha planteado como mejora en evolución de la aplicación.



Al seleccionar cada uno de las diferentes posibilidades del desplegable se actualiza la región 2 con los archivos que cumplan el criterio seleccionado.

- **Región 2:** Muestra el listado de los archivos. Se presentan algunas características importantes del archivo como son el nombre, el tipo de archivo y la extensión. Existe un campo adicional el cual has ido pensando para indicar si el archivo se encuentra almacenado en el propio servidor (local: loc) o está almacenado en un servidor externo⁴⁶.
- **Región 3:** Cuando se selecciona un archivo sobre la región 2, el sistema permite previsualizarlo en la región 3 a través de un reproductor desarrollado para tal fin. Dispone de funcionalidades básicas de reproducción para los vídeos (parar, pausar, reproducir y activar o desactivar el sonido). Estas funcionalidades desaparecen de la interfaz cuando se selecciona una imagen, la cual no dispone de reproducción como cabe esperar.
- **Región 4:** Esta región es bastante dinámica, ya que los botones presentes en esta Zona van cambiando de funcionalidad y nombre según el evento producido en la interfaz.
 - Eliminar Archivo del Servidor: Nos permite eliminar los archivos que no se están utilizando del servidor. Si el archivo se estuviera utilizando en alguna *Ubicación*, éste sería eliminado de la configuración de esa *Ubicación* (de todas las Lista de Reproducción que lo tuvieran programado) y se establecería una orden para que los *ERC* asignados a esa *Ubicación* en cuestión, realizaran una petición de configuración al *ESCC* y por tanto se sincronizaran de forma automática.
 - Subir Archivos al Servidor: Cuando se pulsa sobre este botón nos aparece una pantalla de exploración donde podremos seleccionar diferentes archivos (*.flv, *.jpeg, ...) que tengamos almacenados de forma local en el equipo desde el cual se está realizando la administración del sistema.

⁴⁶ Esta funcionalidad aún no ha sido desarrollada aunque se espera que en futuras implementaciones se provea la posibilidad de programar contenido presente en otro servidor, al cual podríamos denominarlo *Equipo Servidor de Contenido Externo (ESCE)*. El rol de este tipo de servidores simplemente sería la de almacenar el contenido multimedia. Esta idea se mantiene como mejora del *Marco* ante la posibilidad de líneas de trabajo futuro.

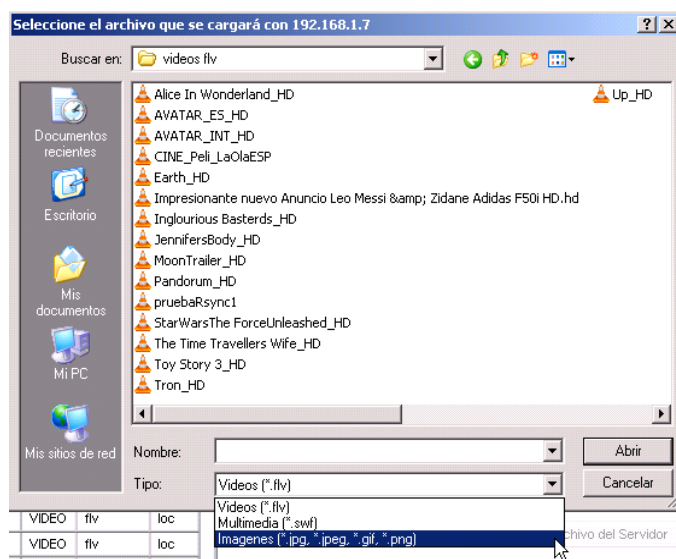


Ilustración 116.-. Diálogo de carga de archivos al servidor.

El tiempo que dura la carga del archivo al servidor aparece una ventana con el estado de la carga mostrando los bytes subidos hasta el momento. Es útil para llevar un seguimiento de la carga del archivo. La carga de archivos al *ESCC* debe ser secuencial y sólo está permitida la carga de un archivo simultáneamente desde el mismo equipo administrador.

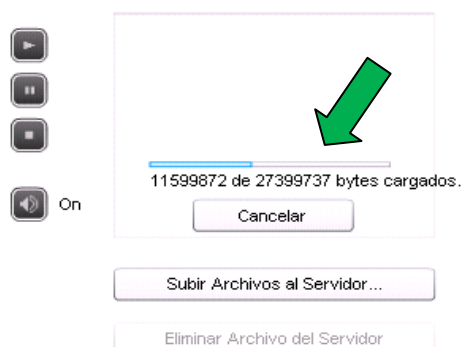
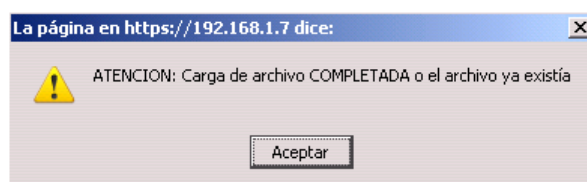


Ilustración 117.-. Barra de progreso de carga del archivo en el servidor.

El sistema además informa de que la carga del archivo se ha llevado a cabo con éxito o se ha sobrescrito el archivo.



ARCHIVOS >> Búsqueda de Archivos

La búsqueda de *archivos* dentro de la programación de las ubicaciones y dentro de las *listas de reproducción* es muy útil a la hora de establecer cómo se ha configurado el sistema.

Las búsquedas que se permiten están enfocadas a mantener una rápida visualización de los vídeos que están programados en las diferentes *ubicaciones*.

A continuación se muestra el resultado de aplicar una búsqueda en una determinada ubicación de una determinada lista:

1 ARCHIVOS cineshd trailers

2 BUSCAR QUITAR

3

ID	Nombre	Tipo	Extensio	Ubicacio	Lista	H Inicio	H Fin
151	Earth_HI	VIDEO	flv	cineshd	trailers	00:00:00	23:00:00
145	MoonTra	VIDEO	flv	cineshd	trailers	00:00:00	23:00:00
143	The Time	VIDEO	flv	cineshd	trailers	00:00:00	23:00:00
141	Tron_HC	VIDEO	flv	cineshd	trailers	00:00:00	23:00:00
140	Up_HD	VIDEO	flv	cineshd	trailers	00:00:00	23:00:00

4

Info Socket:

Se carga http://192.168.1.7:8080/crossdomain.xml
 Conectando...
 Conectado con servidor socket...192.168.1.7
 Datos leídos del S

Info. Socket

Ilustración 118.-. Pantalla de administración para realizar la búsqueda de archivos dentro de las listas asignadas a las ubicaciones.

A continuación se resumen la funcionalidad de cada región presentada:

- **Región 1:** En esta región definida en la Ilustración 118 se permite configurar un filtro para realizar una búsqueda acotada del archivo o de los archivos asociados a una ubicación o lista específica. Se ha de tener en cuenta que las listas con el mismo nombre asociadas a *ubicaciones* diferentes son totalmente independientes ya que corresponden a dos *listas de reproducción* diferentes.

A continuación, se muestran ejemplos del tipo de filtrado que puede realizarse, el cual puede ser buscar un archivo en concreto o presentar los archivos de una lista asociada a una determinada *Ubicación*:

ARCHIVOS

UBICACIONES

LISTAS

Es posible que haya dos listas con el mismo nombre en dos ubicaciones diferentes, éstas corresponderían a dos listas independientes. Sin embargo, en el desplegable de la región 1 sólo aparecerá una sola vez y el resultado ofrecerá los archivos de ambas ubicaciones:

The screenshot shows the 'pfc v1.0 BuscaArchivos' application. At the top is a navigation bar with links: Inicio, UBICACIONES », ARCHIVOS », ACTUALIZAR », SISTEMA », DESCARGAS », MANUAL », and Salir. Below the navigation bar, the title 'pfc v1.0' is displayed in large brown letters, with 'Busqueda de Archivos en la Configuración' in orange below it. To the right, 'BuscaArchivos' is written in large orange letters. The main interface has three dropdown menus: 'ARCHIVOS' (set to 'listaduplicada'), 'UBICACIONES' (set to 'listaduplicada'), and a third dropdown set to 'listaduplicada'. Below these are 'BUSCAR' and 'QUITAR' buttons. A table displays search results:

ID	Nombre	Tipo	Extensión	Ubicación	Lista	H Inicial	H Final
202	motoEsti	VIDEO	flv	cineshd	listaduplicada	00:0	00:0
162	cine_cla	IMAG	jpg	cultCien	listaduplicada	00:0	00:0

The 'Ubicación' and 'Lista' columns for the second row are highlighted with a red box. To the right of the table is a large empty rectangular box. At the bottom right, there is an 'Info Socket:' section with a log showing connection status: 'Se carga http://192.168.103.7:8080/crossdomain.xml', 'Conectando...', 'Conectado con servidor socket...192.168.103.7', and 'Datos leídos del Servidor:'.

Ilustración 119.-. Resultado de buscar una lista con nombre duplicado en el sistema.

- **Región 2:** La región 2 nos permite seleccionar la acción a llevar a cabo. Por una parte, se puede realizar una búsqueda según el filtro aplicado en la región 1 o bien, seleccionando un archivo, quitarlo de la configuración.

Se ha de tener en cuenta que cuando se quita un archivo de la configuración se enviarán órdenes a las *ubicaciones* implicadas; esto es una forma rápida de poder gestionar las *listas de reproducción* asociadas a las *ubicaciones*. Cuando se elimina un archivo pulsando el botón 'Quitar' se ha de notar que el archivo no desaparece del sistema sólo de la configuración asociada.

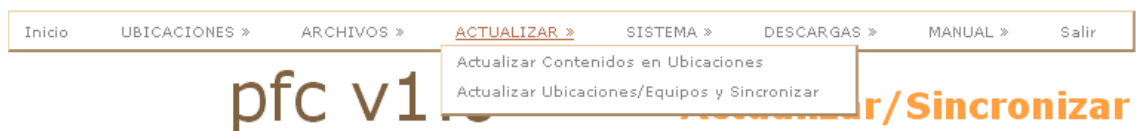
- **Región 3:** La región 3 muestra un listado de los archivos encontrados de acuerdo al criterio de búsqueda de archivos. Estas regiones muestran información acerca del archivo en la configuración y un mismo archivo puede aparecer repetido varias veces. La región 3 nos indica parámetros relacionados con la *lista de reproducción* en el que ha sido programado.
- **Región 4:** La región 4 nos muestra el archivo con la funcionalidad de poder previsualizarlo. Para ello se ha de haber seleccionado uno de los archivos listados en la región 3. Los botones son los mismos que en el resto de pantallas donde se permite la previsualización de los archivos del sistema.

- **Región Info. Socket:** Muestra información del *socket*. Una vez que la aplicación se ha conectado nos muestra la información de las órdenes que se han enviado a través del *socket*.

C.I.3. MENÚ ACTUALIZAR

El menú ACTUALIZAR, es importante ya que es la interfaz que nos permite enviar órdenes a los diferentes *ERCs*. Estas órdenes se podrán realizar o bien a *ubicaciones* en general, a *ERC* directamente u órdenes a que se aplican al sistema en general. El menú 'ACTUALIZAR' se ha concebido, por tanto, para enviar órdenes directamente a los *ERCs* desde la *Interfaz Gráfica de Usuario* que configura el *ESCC*. De este modo si algún equipo no hubiera recibido una orden de sincronización por algún problema de conectividad de los equipos que integran el *Marco*, ésta podría enviarse de nuevo a dichos equipos de forma explícita.

Se compone de dos submenús, uno para enviar ordenes de actualización al conjunto de *ERCs* que conforman una *Ubicación* ('Actualizar Contenidos de Ubicaciones') y otra pantalla que permite una mayor flexibilidad y que incluso permite enviar órdenes a determinados equipos de forma única o incluso enviar la orden de sincronización de contenido multimedia ('Actualizar Ubicaciones/ Equipos y Sincronizar').



ACTUALIZAR >> Contenidos en Ubicaciones

Desde esta pantalla se permite indicar a los *ERCs*, que pertenecen a una determinada *Ubicación*, que realicen una nueva petición de la configuración.

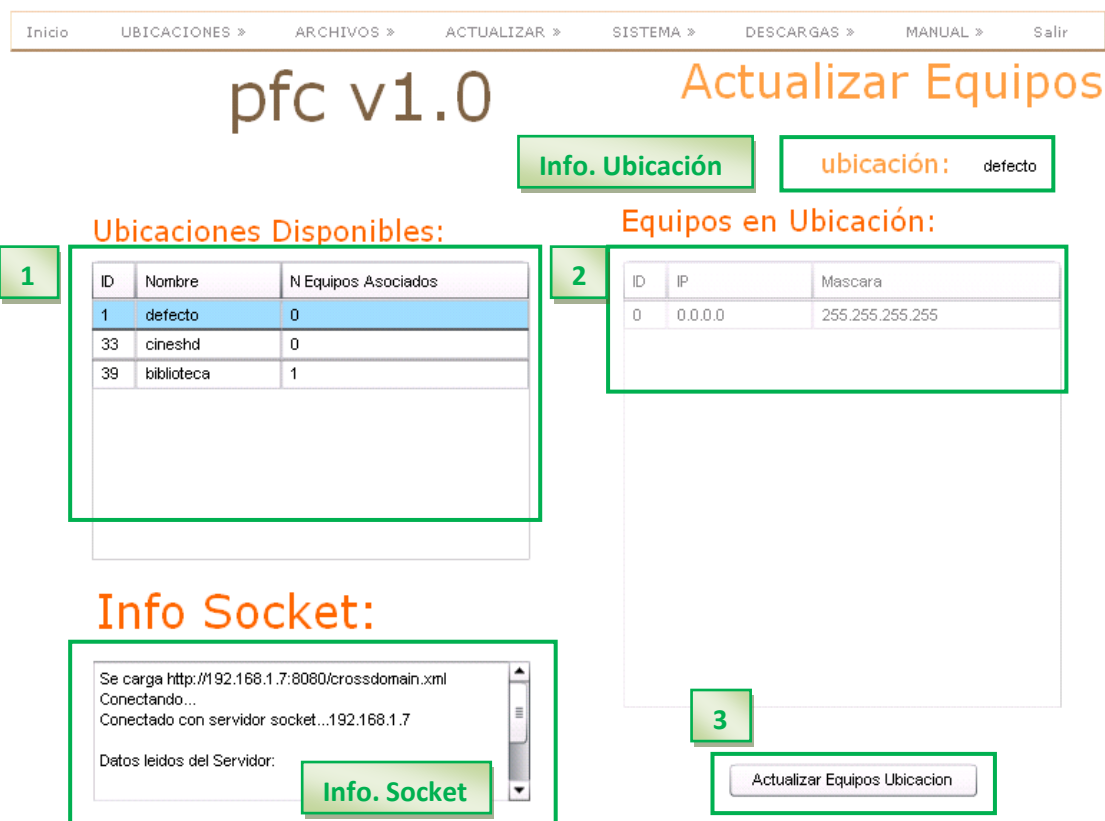


Ilustración 120.-. Pantalla de administración para enviar ordenes de actualización a todo el conjunto de equipos suscritos a una Ubicación.

La funcionalidad de las diferentes regiones es la siguiente:

- **Región Info. Ubicación:** La funcionalidad de este panel es la misma que en el resto de pantallas donde aparece. Nos informa de la *Ubicación* que se está configurando. La selección de la *Ubicación* que se está configurando se realiza desde la pantalla 'UBICACIONES >> Seleccionar/Crear Ubicación'. En este caso no es un valor determinante para la configuración de la pantalla como en otros casos, debido a que se permite enviar la orden a los equipos de cualquiera de las *ubicaciones* configuradas.
- **Región 1:** Esta región muestra el listado de *ubicaciones* configuradas en el sistema e informa del número de *ERCs* asociados a una determinada *Ubicación*. Una vez se pulsa sobre una *Ubicación* de este listado, la región 2 actualiza sus datos informando la *dirección IPv4* de los *ERCs* asociados. De esta forma se tiene un resumen del número de equipos a los que afectaría el envío de la orden.
- **Región 2:** Esta región muestra la *dirección IPv4* de los *ERCs* asociado a la *Ubicación* que se ha seleccionado en la región 1. Cuando se envíe la orden afectará a todos los equipos que se muestren en el listado de esta región.
- **Región 3:** Pulsando sobre el botón de esta región ('Actualizar Equipos Ubicación') se envía una orden a través del *servidor socket* para que todos los *ERCs* asociados a la *Ubicación* que se muestran en la región 2 realicen una petición al *ESCC* para obtener la configuración de la *Ubicación* a la que pertenecen.
- **Región Info. Socket:** Al igual que en el resto de pantallas esta región permite visualizar las órdenes que se están enviando mediante el *servidor socket*.

ACTUALIZAR >> Actualizar Ubicaciones/Equipos y Sincronizar

Permite enviar otras órdenes diferentes a la presentada en el apartado anterior. Permite que la orden de actualización de la configuración vaya dirigida a un solo equipo y no a todos los equipos de una *Ubicación*. Además se introduce la orden ‘sincronizar’ que indica a los *ERCs* que deben recuperar los archivos del *ESCC* y almacenarlos de forma local sincronizando el contenido multimedia.

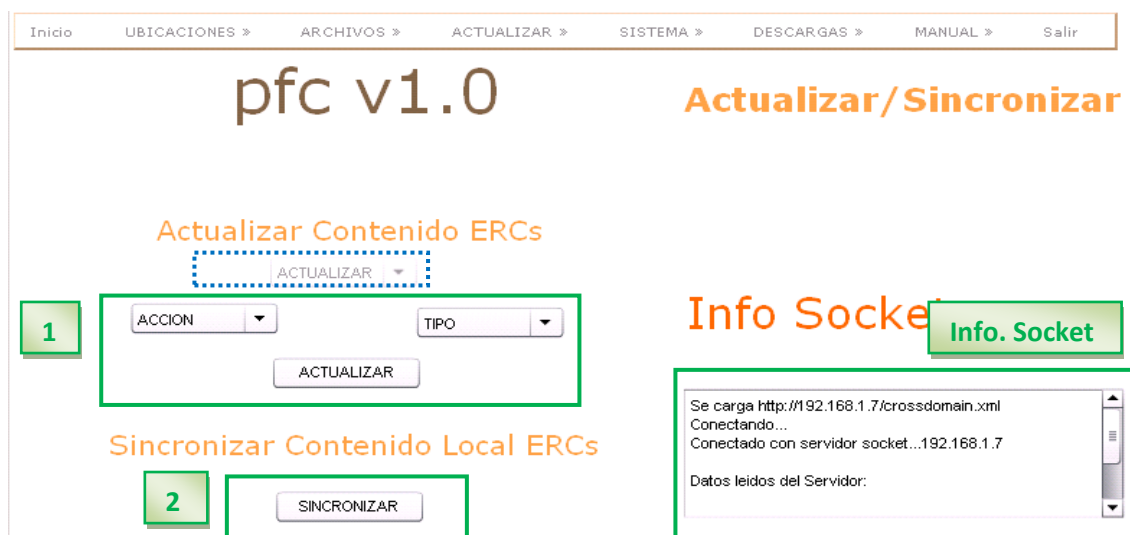
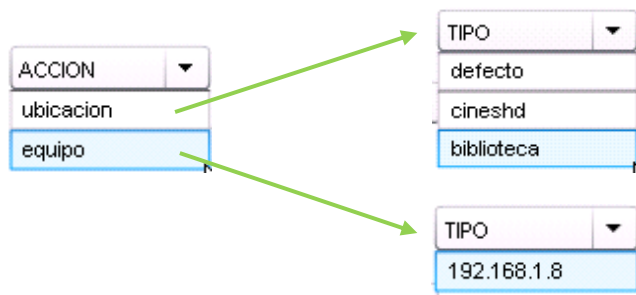


Ilustración 121.-. Pantalla de gestión de actualización peticiones y ordenes de sincronismo de contenido.

- **Región 1:** Esta región permite seleccionar el tipo de actualización que se quiere realizar en el sistema.
 - Se puede establecer una orden para que se realicen peticiones de configuración desde los *ERCs* al *ESCC* basadas en el *Ubicación* (desplegable *ACCION*= '*ubicación*'). De este modo todos los *ERCs* pertenecientes a la *Ubicación* realizarán nuevas peticiones y se actualizarán a la configuración que se ha establecido en el *ESCC* en ese instante.
 - Además, también se ha habilitado la funcionalidad que se puede indicar a un *ERC* (desplegable *ACCION* = '*equipo*') que realice una petición de actualización de configuración de forma aislada.

A continuación, se muestra un ejemplo del desplegable *TIPO* que se muestra en la Ilustración 121, según se realice una acción u otra (desplegable '*ACCION*')

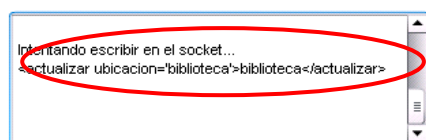


- **Región 2:** Permite establecer la orden de sincronización de forma general. Así, cuando se envíe esta orden de forma manual, todos los *ERC* que la reciban

comenzarán a sincronizar su contenido local con el del *ESCC*. Este sincronismo se realiza con un determinado *ancho de banda* el cual se configura en cada *ERC* a través de una interfaz web y que se detallará en el apéndice C.II (CONFIGURACION >> Ancho de Banda Sincro).

- **Región Info. Socket:** En esta región se muestra la información relativa al *socket*. Se verifica que existe conexión y que la orden efectivamente se ha escrito en el *socket*.

Info Socket:



Info Socket:

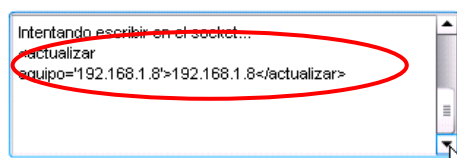


Ilustración 122.-. Información escrita en el socket cuando se actualiza una ubicación (izquierda) y cuando se indica que actualice un sólo ERC (derecha).

La ilustración anterior muestra dos ejemplos de órdenes que pueden darse en función de la elección seleccionada en el desplegable ACCION ('ubicación' o 'equipo')

C.I.4. MENÚ SISTEMA

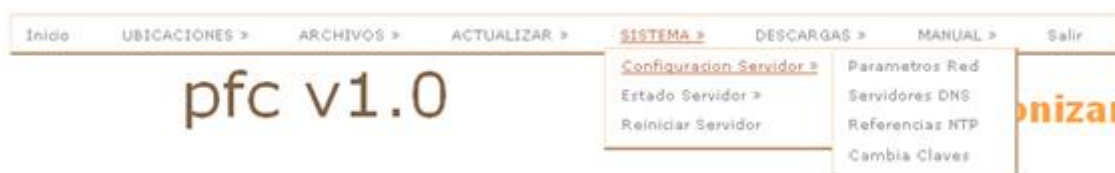
Son tres acciones básicas las que se pueden realizar dentro del menú SISTEMA:

- Configurar parámetros relativos al *Sistema Operativo*
- Verificar ciertos aspectos del estado del *ESCC* relacionados con los parámetros del sistema
- Reiniciar el *ESCC*

El menú SISTEMA y las páginas a las cuales se acceden sólo son accesibles mediante el usuario '*sistema*', quedando el acceso restringido para el usuario '*admin*'.

A continuación se muestran cada una de las pantallas de configuración de los parámetros del sistema del *ESCC*:

SISTEMA >> Configuración Servidor



El menú 'SISTEMA>> Configuración de Servidor' permite el acceso a la edición de varios parámetros del sistema:

- Parámetros de Red
- Servidores DNS
- Referencias NTP

- Cambio de Claves de los Usuarios

En los sucesivos apartados se muestra la pantalla a la que se tiene acceso para realizar las modificaciones de los diferentes parámetros:

SISTEMA >> Configuración Servidor >> Parámetros de Red

Inicio UBICACIONES > ARCHIVOS > ACTUALIZAR > SISTEMA > DESCARGAS > MANUAL > Salir

Config. Red Servidor

Interfaz VMWARE: eth0

IP: 192 . 168 . 103 . 7

MASCARA: 255 . 255 . 255 . 0

GW: 192 . 168 . 103 . 1

EDITAR ACTUAL

Ilustración 123.-. Pantalla de Edición de los parámetros de red del ESCC.

En esta pantalla se permite modificar los parámetros de red editando las áreas de texto donde se muestran dichos parámetros. El botón 'EDITAR' permite cambiar la configuración y el botón 'ACTUAL' rellena las áreas de texto con los valores actuales almacenados en la base de datos.

La modificación de estos parámetros no se aplica de forma inmediata, sino que se habilita una sub interfaz con estos valores de modo que no se pierde la conectividad. Posteriormente para que el sistema disponga de esta nueva configuración y pierda la anterior se ha de reiniciar el sistema.

SISTEMA >> Configuración Servidor >> Servidores DNS

Resulta de interés poder modificar los servidores DNS que utilizará el equipo ESCC. Esto es posible a través de esta pantalla de edición:

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

Config. Cliente DNS

DNS Primario:

DNS Secundario:

Ilustración 124.-. Edición de los servidores DNS en el ESCC.

El sistema permite editar las direcciones IPv4 de dos servidores *DNS*. Es útil establecer servidores *DNS* ante la posibilidad de configurar las referencias *NTP* de diferentes dominios.

SISTEMA >> Configuración Servidor >> Referencias NTP

El sistema permite establecer referencias *NTP* que permitirán sincronizar la hora del *ESCC* con dichos servidores.

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

Referencias NTP del Servidor

Ref. NTP 1:

Ref. NTP 2:

Ref. NTP 3:

Ilustración 125.-. Edición de las referencias NTP en el ESCC.

Las referencias pueden introducirse tanto como nombres de dominios, como con la propia dirección IPv4. Es posible introducir varias referencias *NTP*. El funcionamiento de los botones 'EDITAR' y 'ACTUAL' es el mismo que en el resto de pantallas de edición de parámetros en el servidor.

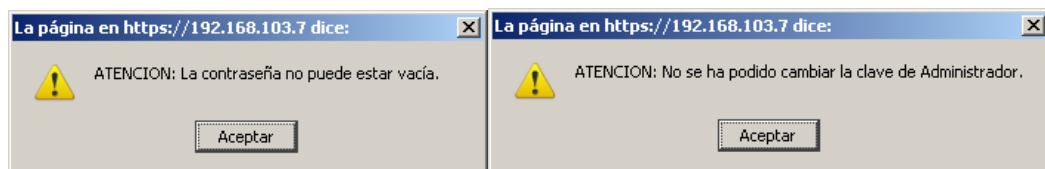
SISTEMA >> Configuración Servidor >> Cambia Claves

La siguiente pantalla de configuración de parámetros del servidor permite relaizar un cambio de las claves de los usuarios del sistema:

Ilustración 126.- Pantalla de Edición de cambio de contraseña para los usuarios 'admin' y 'sistema'.

En función de la selección del desplegable se permitirá cambiar la contraseña al usuario 'admin' o el usuario 'sistema'.

El sistema es capaz de identificar una serie de errores al intentar modificar la clave:



Para poder realizar el cambio de claves se ha de conocer la contraseña del usuario que se quiera modificar. La nueva contraseña se debe introducir dos veces y para que el proceso se lleve a cabo deben coincidir al pulsar el botón 'ACTUALIZAR'.

SISTEMA >> Estado Servidor

A continuación se muestran una serie de pantallas que nos permiten comprobar el estado del servidor. Estas pantallas son de gran utilidad a la hora de establecer una estrategia de resolución de problemas.



Se pueden observar el estado de la configuración de red, el estado de los servidores DNS y las referencias NTP.

SISTEMA >> Estado Servidor >> Estado Red

La siguiente imagen muestra un ejemplo de la pantalla devuelta al pulsar sobre este enlace:

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

Estado:ifconfig

```

eth0      Link encap:Ethernet  HWaddr 00:0c:29:c4:61:e4
          inet addr:192.168.1.7  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec4:61e4/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:388382 errors:0 dropped:0 overruns:0 frame:0
          TX packets:463030 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:178638310 (178.6 MB)  TX bytes:608242896 (608.2 MB)
          Interrupt:18 Base address:0x1400

eth0:0    Link encap:Ethernet  HWaddr 00:0c:29:c4:61:e4
          inet addr:192.168.69.2  Bcast:192.168.69.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:18 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:739 errors:0 dropped:0 overruns:0 frame:0
          TX packets:739 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:48936 (48.9 KB)  TX bytes:48936 (48.9 KB)

```

Estado:route

```

Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.69.0     *              255.255.255.0   U        0      0        0 eth0
192.168.1.0      *              255.255.255.0   U        0      0        0 eth0
default          192.168.1.1    0.0.0.0         UG       100    0        0 eth0

```

[Última Modificación](#)

Ilustración 127.-. Pantalla que muestra el estado de los parámetros de red .

Cuando se pulsa sobre el hipervínculo 'Última Modificación' accedemos a una nueva pantalla y se presenta la información almacenada en la última modificación que se realizó en el sistema. Un ejemplo de la información devuelta se muestra a continuación:

```

IP Origen: 192.168.103.15

IP Configurada: 192.168.103.7

* Stopping NTP server ntpd
...done.
* Starting NTP server ntpd
...done.

PING 192.168.103.1 (192.168.103.1) from 192.168.103.7 : 56(84) bytes of
data.
From 192.168.103.7 icmp_seq=1 Destination Host Unreachable
From 192.168.103.7 icmp_seq=2 Destination Host Unreachable

--- 192.168.103.1 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1000ms
, pipe 2

PING 192.168.103.15 (192.168.103.15) from 192.168.103.7 : 56(84) bytes of
data.
64 bytes from 192.168.103.15: icmp_seq=1 ttl=128 time=0.185 ms
64 bytes from 192.168.103.15: icmp_seq=2 ttl=128 time=0.157 ms

```



```
--- 192.168.103.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.157/0.171/0.185/0.014 ms
```

IP Por defecto: 169.254.69.2

Esta información se presenta sobre una nueva pantalla no variando la navegación del usuario que está supervisando el sistema. Cuando se realiza un cambio en la configuración se establece un ping desde el ESCC hacia el router por defecto configurado y hacia el equipo desde el que se ha realizado la configuración. Esta información se almacena en un fichero y es la que se presenta en una página independiente.

SISTEMA >> Estado Servidor >> Estado DNS

Esta pantalla permite verificar el estado de los servidores DNS configurados:

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

Estado: /etc/resolv.conf

```
nameserver 192.168.103.15
nameserver 192.168.103.1
```

PING:

```
PING 192.168.103.15 (192.168.103.15) 56(84) bytes of data.
64 bytes from 192.168.103.15: icmp_seq=1 ttl=128 time=0.167 ms
64 bytes from 192.168.103.15: icmp_seq=2 ttl=128 time=0.165 ms

--- 192.168.103.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.165/0.166/0.167/0.001 ms
PING 192.168.103.1 (192.168.103.1) 56(84) bytes of data.
From 192.168.103.7 icmp_seq=1 Destination Host Unreachable
From 192.168.103.7 icmp_seq=2 Destination Host Unreachable

--- 192.168.103.1 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 999ms
, pipe 2
```

Ilustración 128.-. Resultado de verificar la configuración de los servidores DNS y la conectividad a estos mediante ping.

Inicialmente se muestra el archivo de configuración de los DNS y posteriormente se realiza un ping de dos peticiones a cada uno de los servidores. En el ejemplo anterior se muestra que el primero de los servidores configurado responde a las peticiones, mientras que el segundo parece inactivo (según PING).

SISTEMA >> Estado Servidor >> Estado NTP

La siguiente pantalla muestra un ejemplo del estado de los servidores NTP configurados:

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

```

Estado: /etc/ntp.conf

driftfile /var/lib/ntp/ntp.drift
server 192.168.103.15
server ntp.ubuntu.com
server 127.127.1.1 minpoll 4
fudge 127.127.1.1 stratum 10

Estado: date

Mon Dec  6 03:25:56 CET 2010

Estado: host

sudo host 192.168.103.15

;; connection timed out; no servers could be reached

sudo host ntp.ubuntu.com

;; connection timed out; no servers could be reached

sudo host

Estado: ping

PING 192.168.103.15 (192.168.103.15) 56(84) bytes of data.
64 bytes from 192.168.103.15: icmp_seq=1 ttl=128 time=0.169 ms
64 bytes from 192.168.103.15: icmp_seq=2 ttl=128 time=0.118 ms

--- 192.168.103.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.118/0.143/0.169/0.028 ms

```

Ilustración 129.-. Estado de los tres servidores NTP configurados.

La pantalla muestras pequeñas pruebas que se realizan sobre los servidores configurados. El objetivo es determinar algún problema con la conectividad a dichos servidores.

SISTEMA >> Reiniciar

Al pulsar dicho enlace el servidor comenzará con el proceso de reinicio:

Inicio	UBICACIONES »	ARCHIVOS »	ACTUALIZAR »	SISTEMA »	DESCARGAS »	MANUAL »	Salir
--------	---------------	------------	--------------	-----------	-------------	----------	-------

Reiniciando...

C.I.5. MENÚ DESCARGAS

Se han proporcionado enlaces a aplicaciones software de interés tanto para el *Sistema Operativo Ubuntu*, como para *Windows*.



DESCARGAS >> Enlaces Software

Son enlaces a dos aplicaciones que se han considerado que pueden ser útiles a la hora de administrar y configurar la aplicación de *Difusión Selectiva*. A continuación se muestran los enlaces a ambas pantallas:



Ilustración 130.-. Hiperenlaces a programas útiles para administrar el sistema.

Estas aplicaciones tienen las siguientes funciones:

- **Adobe Flash Player:** Necesaria para la previsualización de contenido multimedia en las ubicaciones y la configuración del sistema a través de la *Interfaz Gráfica*.
- **FreeVideConverter/Win FF Converter:** Estos programas se han seleccionado para proporcionar un conversor del *códec* de vídeo a los *codecs* correspondientes permitidos para el contenedor multimedia *flv*. A partir de otros formatos de vídeo (contenedores multimedia) estos programas son capaces de generar archivos 'flv' (Flash Video), el cual es el tipo de archivo de vídeo que es capaz de programar y reproducir la aplicación desarrollada.

DESCARGAS >> Archivos

Se permite la descarga directa de los archivos subidos al servidor. El menú muestra los enlaces directos a un tipo de búsqueda concreta. Así pulsando sobre DESCARGAS >> Archivos >> Imágenes, sólo se listarán los enlaces directos a las imágenes cargadas en el ESCC. Así, se establece el mismo comportamiento para el resto de tipos de archivos.



A continuación se muestra una imagen de la pantalla que se presenta cuando se listan los archivos presentes en el servidor:

Archivos en ESCC

1

☐ Videos ☒ Imágenes ☐ swf (as3)

BUSCAR

Espacio libre: 5 GB 166 MB

2

Archivo	KB	Tipo
long300x250_01.jpg	12	IMAGENES
Thumbs.db	101	IMAGENES
benj800x250a_01.jpg	42	IMAGENES
5.png	18	IMAGENES
benj300x250.jpg	21	IMAGENES
300x250a_01.jpg	17	IMAGENES
Imagen081.jpg	23	IMAGENES
benj800x250_01.jpg	40	IMAGENES
benj800x250c_01.jpg	36	IMAGENES
fotolog2.jpg	83	IMAGENES
Imagen45.jpg	117	IMAGENES
long160x600.jpg	15	IMAGENES

Ilustración 131.-Listado de Archivos Multimedia almacenados en el ESCC.

El listado de archivos se presenta con enlaces a estos permitiendo su descarga. Adicionalmente se presenta el tamaño que ocupan en disco y el espacio disponible en el servidor. Los enlaces que aparecen en el menú son dirigidos a la misma página pero se les proporciona ciertas opciones (parámetros GET) indicando que tipo de archivos se quieren listar.

A continuación se presenta una pequeña explicación de cada región establecida en la Ilustración 131:

- **Región 1:** Muestra las opciones de búsqueda de los archivos presentes en el servidor. Se presenta la posibilidad de listar los archivos *swf* que se hayan subido al servidor.

NOTA: Actualmente no se ha proporcionado la funcionalidad de utilizar el tipo de archivo SWF debido a la incompatibilidad de versiones detectada. Se ha proporcionado el Marco en la arquitectura para que sean cargados en el ESCC; sin embargo, no se ha provisto su programación y reproducción en los diferentes Escenarios. Se espera que en versiones posteriores del software se prevea esta funcionalidad.

- **Región 2:** Se muestran los archivos almacenados en el ESCC. Cada elemento de la lista es un enlace a la descarga del archivo que se abre en una nueva ventana. Se muestra información adicional del tipo de archivo y el tamaño que ocupan en disco.

C.II. Manual del Usuario Administrador del ERC

La administración del ERC se realiza en varios pasos sencillos. Las opciones configurables de este equipo son básicamente 3:

- Establecer los parámetros de direccionamiento de red.
- Indicar cuál es el direccionamiento IPv4 del ESCC.
- Establecer el ancho de banda a utilizar en la sincronización del contenido multimedia con el ESCC.

A continuación se muestra de forma secuencial las configuraciones que se deben realizar en el cada equipo:

A través del navegador web, accedemos a la dirección IP del equipo que ese haya configurado anteriormente o accedemos a través de un cable cruzado conectado directamente desde otro equipo a la dirección 169.254.69.2. De este modo nos aparece en el navegador la siguiente pantalla de acceso:



Ilustración 132.-. Interfaz de Acceso al ERC.

Como se puede apreciar, en la barra de direcciones el protocolo utilizado para acceder es seguro (*https*). Por tanto, se ha de aceptar un certificado de seguridad previo para la carga de la página de configuración.

Una vez se introduce el usuario (*sistema* o *admin*) y la contraseña, accedemos a la siguiente pantalla de presentación:



Dependiendo de con qué usuario se acceda al sistema se tendrá un menú de configuración u otro en la página de administración:

- **sistema**



- **admin**



Como se observa, el usuario *admin* no tendrá acceso al menú SISTEMA, el cual da acceso a las páginas de configuración del sistema en las cuales se configura el *direccionamiento IPv4* del ERC, *máscaras* y *router por defecto*, y las páginas que permite configurar las claves de los usuarios. Se ha de hacer indicar que el acceso restringido no sólo es a los enlaces del menú, sino que también se restringe el acceso a la propia página.

C.II.1. MENU CONFIGURACIÓN

A continuación se muestra el menú CONFIGURACIÓN, el cual muestra dos opciones a configurar:

- *Direccionamiento IPv4* del ESCC.
- Ancho de banda Utilizado para la sincronización del contenido multimedia con el ESCC.



Configuración de Ingeniería Telemática

CONFIGURACION >> Direccionamiento IP ESCC

Con la pestaña CONFIGURACION >> Direccionamiento IP ESCC tendremos acceso a la página web donde se puede configurar la *dirección IPv4* del ESCC. Como veremos posteriormente en el menú *ESTADO*, podremos verificar la conectividad a este equipo, lo cual nos proporcionará información de interés a la hora de resolver problemas en la configuración del sistema.



Config. IP del ESCC

IP Serv:

IP BackUp:

EDITAR

RESET

[PING a los servidores](#)

Ilustración 133.-. Configuración de la Dirección IPv4 del ESCC del sistema al que pertenece el ERC.

La segunda *dirección IPv4*, a la que se ha denominado *IP BackUP*, tiene como propósito ser utilizada para establecer un sistema de redundancia, de modo que podría obtenerse la configuración de este segundo equipo si el sistema fallara. Actualmente esta *dirección IPv4* no es utilizada por el equipo *ERC*.

CONFIGURACION >> Ancho de Banda Sincro

En esta pantalla se configura en el equipo el *ancho de banda* utilizado para obtener una copia en local del contenido multimedia. Así, se recuperan desde el servidor los archivos multimedia que se han subido al ESCC desde el interfaz de administrador de sistema. Para ello, marcamos en *K Bytes/seg* el *ancho de banda* a utilizar. Este parámetro es importante a la hora de no saturar el ancho de banda disponible desde el ESCC al *ERC*.

A nivel de lógica en el *Marco*, se ha elegido que la descarga del contenido multimedia desde el *ESCC* se realice en cuatro situaciones:

- Cuando se edita el valor del ancho de banda de sincronismo en el ERC (pulsar botón EDITAR).
- Cuando se da la orden de sincronización en el ERC (pulsar botón SINCRO).
- Cuando se inicia o reinicia el ERC.
- Cuando se sube o elimina un archivo al ESCC.

A continuación se presenta la pantalla de edición del parámetro *ancho de banda* de sincronización:



Ilustración 134.-. Configuración del Ancho de Banda utilizado para sincronizar el contenido multimedia desde el *ESCC* al *ERC*.

El botón RESET permite establece en la pantalla el valor que tiene almacenado actualmente.

C.II.2. MENÚ SISTEMA

En el menú SISTEMA se permite el acceso a los parámetros de configuración del sistema y las acciones de mantenimiento como, por ejemplo, ser la posibilidad de reiniciar el sistema. Estos parámetros sólo pueden ser modificados por el usuario '*sistema*'.



Ilustración 135.-. Menú SISTEMA.

Las opciones que se proveen en este menú son:

- Edición del Direccionamiento de red del ERC.
- Edición Clave de los usuarios sistema y admin.
- Reiniciar el equipo.

SISTEMA >> Direccionamiento IP ERC

Dentro del menú SISTEMA >> Direccionamiento IP ERC, accedemos a la siguiente pantalla de configuración:

Inicio CONFIGURACION > SISTEMA > ESTADO > Salir

Config. Red ERC

Interfaz VMWARE:

IP:

MASCARA:

GW:

Ilustración 136.-. Página de configuración del direccionamiento IPv4 del ERC.

Los parámetros configurables son la *dirección IPv4*, la *máscara* de red y el GW (*router por defecto* o *puerta de enlace*). Una vez que se editan estos parámetros se mantiene los anteriores en la configuración del sistema hasta que se produce el reinicio. Esto se ha decidido así debido a que se podría necesitar realizar alguna otra configuración de modo que no se quiere perder la conectividad con las páginas de configuración. Una vez se reinicia el sistema se pierde la configuración anterior y se carga, desde el registro de la *tabla* de la *base de datos* que almacena la configuración, la configuración establecida a través de esta página de configuración.

SISTEMA >> Clave Admin/Clave Sistema

A través del menú SISTEMA >> Clave Admin/Clave Sistema accedemos a las pantallas en las que podemos variar las claves de acceso de los usuarios que tienen acceso a los diferentes menús de configuración del ERC.

Ilustración 137.- Acceso al cambio de claves de los usuarios definidos en el sistema.

Se establecen las claves por defecto '1234' en ambos usuarios como claves por defecto. Sin embargo, el botón RESET reinicia los valores del formulario a valores vacíos sin establecer la clave a '1234'.

C.II.3. MENÚ ESTADO

El último elemento representativo del menú proporciona a los usuarios información acerca del estado del sistema. Está compuesto por tres elementos que ayudan a establecer el estado del sistema y también a obtener información ante una posible estrategia de resolución de problemas si los hubiera.



Ilustración 138.- Menú ESTADO del ERC.

ESTADO >> Estado Red Cliente

Se puede diagnosticar el estado de la red del ERC, pulsando en el menú sobre ESTADO >> Estado Red Cliente. El sistema ejecutará una serie de instrucciones sencillas que mostrarán la configuración del sistema. Un ejemplo de los resultados obtenido se muestra en la siguiente ilustración:

Inicio	CONFIGURACION >	SISTEMA >	ESTADO >	Salir
--------	-----------------	-----------	----------	-------



Estado Sistema

Estado:ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:4c:2a:eb
          inet addr:192.168.1.8  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe4c:2aeb/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:780 errors:0 dropped:0 overruns:0 frame:0
          TX packets:896 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:128759 (128.7 KB)  TX bytes:170219 (170.2 KB)
          Interrupt:18 Base address:0x1400

eth0:0    Link encap:Ethernet  HWaddr 00:0c:29:4c:2a:eb
          inet addr:192.168.69.2  Bcast:192.168.69.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:18 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:209 errors:0 dropped:0 overruns:0 frame:0
          TX packets:209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:190849 (190.8 KB)  TX bytes:190849 (190.8 KB)
```

Estado:route

```
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.69.0   *               255.255.255.0   U        0      0        0 eth0
192.168.1.0    *               255.255.255.0   U        0      0        0 eth0
link-local     *               255.255.0.0     U       1000    0        0 eth0
default        192.168.1.1     0.0.0.0         UG       100    0        0 eth0
```

Estado: DNS /etc/resolv.conf

[Ultima Modificacion](#)

Ilustración 139.-. Estado de los parámetros de red del ERC.

Se observa de forma resumida las configuraciones establecidas en el *ERC* a nivel de *sistema operativo*.

ESTADO >> PING Servidores

Pulsando en el enlace del menú ESTADO >> PING Servidores, se realiza un ping a los distintos equipos relevantes en la arquitectura. Esto provee un mecanismo sencillo de diagnostico para comprobar la conectividad en el *Marco* a nivel de red.

A continuación se muestra un ejemplo del resultado obtenido desde la interfaz gráfica de usuario cuando hay conectividad y cuando no la hay (por ejemplo, equipo *ESCC* apagado):

Inicio	CONFIGURACION »	SISTEMA »	ESTADO »	Salir
--------	-----------------	-----------	----------	-------

PING Servidores...desde 192.168.1.8:

```
PING 192.168.1.7 (192.168.1.7) from 192.168.1.8 : 56(84) bytes of data.
64 bytes from 192.168.1.7: icmp_seq=1 ttl=64 time=0.497 ms
64 bytes from 192.168.1.7: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 192.168.1.7: icmp_seq=3 ttl=64 time=0.245 ms
```

```
--- 192.168.1.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.245/0.347/0.497/0.108 ms
```

```
PING 192.168.1.1 (192.168.1.1) from 192.168.1.8 : 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1092 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=93.5 ms
```

```
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 93.525/593.244/1092.963/499.719 ms, pipe 2
```

```
ping -I 192.168.1.8 -c 2 192.168.1.1
```

[PING a los servidores](#)

Ilustración 140.-. Respuesta del ESCC ante un ping desde el ERC cuando existe conectividad.

Cuando no se tiene conectividad desde el *ERC* al *ESCC*, y no existe elemento de red que descarten el tráfico *ICMP* como pudieran ser firewalls, el mensaje que se presenta en la pantalla de estado sería similar a:

Inicio	CONFIGURACION »	SISTEMA »	ESTADO »	Salir
--------	-----------------	-----------	----------	-------

PING Servidores...desde 192.168.1.8:

```
PING 192.168.1.7 (192.168.1.7) from 192.168.1.8 : 56(84) bytes of data.
From 192.168.1.8 icmp_seq=1 Destination Host Unreachable
From 192.168.1.8 icmp_seq=2 Destination Host Unreachable
From 192.168.1.8 icmp_seq=3 Destination Host Unreachable
```

```
--- 192.168.1.7 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2017ms
, pipe 3
```

```
PING 192.168.1.1 (192.168.1.1) from 192.168.1.8 : 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=5.36 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.58 ms
```

```
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.581/3.472/5.363/1.891 ms
```

```
sudo sudo ping -I 192.168.1.8 -c 2 192.168.1.1
```

[PING a los servidores](#)

Ilustración 141.-. Respuesta del ESCC ante un ping desde el ERC cuando no existe conectividad.

En este caso se realiza *ping* al servidor principal (192.168.1.8) y *ping* al servidor de BackUp (192.168.1.1). Como sólo responde el 192.168.1.1, y están los dos equipos en el mismo

segmento de red pudiera ser que existen problemas en la configuración del ESCC o que estuviera apagado. No obstante, esta prueba se debería realizar como primer diagnóstico, ya que si el ping fuera exitoso y el sistema no respondiera como se espera pudiera ser la causa de error otros elementos como son firewalls presentes en la red. Ya que son varios los puertos en el servidor que deben estar abiertos (8080, 8081, 843) desde el ERC para que el sistema funcione correctamente.

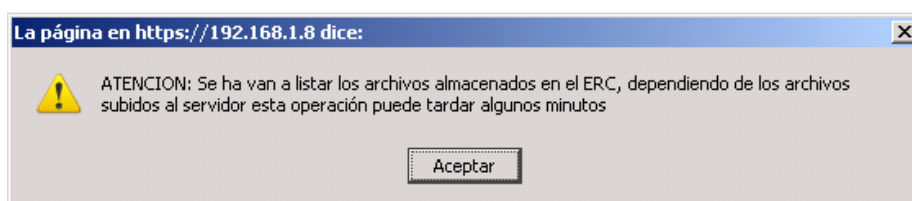
ESTADO >> Archivos Locales

Dentro del menú ESTADO >> Archivos Locales, se puede consultar la lista de archivos que el ERC tiene sincronizado con el ESCC. La pantalla de presentación es la siguiente:



Ilustración 142.-. Acceso a la búsqueda de archivos multimedia almacenados en el ERC.

Se puede elegir el tipo de archivos a consultar y aparece también el espacio libre que queda en el disco duro del equipo. Aparece un mensaje de cortesía en el que se indica que podría tardar varios minutos dependiendo del número de archivos almacenados en el sistema; sin embargo, el resultado aparece de forma inmediata si se tiene una lista de archivos moderada.



Una vez se ejerce la búsqueda, se proporciona un listado a modo de enlace de los archivos multimedia encontrados. Así pulsando sobre estos enlaces se puede realizar una descarga del archivo. En la siguiente ilustración se muestra un ejemplo de una búsqueda realizada en el entorno de pruebas:

[Inicio](#) [CONFIGURACIÓN >](#) [SISTEMA >](#) [ESTADO >](#) [Salir](#)

ERC

☐ Videos ☒ Imágenes ☐ swf (as3)

BUSCAR

Espacio libre: 4 GB

Archivo	KB	Tipo
Imagen45.jpg	117	IMAGENES
5.png	18	IMAGENES
fotolog2.jpg	83	IMAGENES
benj800x250_01.jpg	40	IMAGENES
benj300x250.jpg	21	IMAGENES
benj800x250a_01.jpg	42	IMAGENES
long160x600.jpg	15	IMAGENES
long300x250_01.jpg	12	IMAGENES
300x250a_01.jpg	17	IMAGENES
benj800x250c_01.jpg	36	IMAGENES
Imagen081.jpg	23	IMAGENES

Ilustración 143.-. Listado de Imágenes descargadas de forma local en un equipo ERC.

Se puede indicar que además de indicar el tipo de archivo se muestra información el espacio que ocupa en disco.

D Instalación de Media Wiki en el Sistema

Para proveer ayuda al administrador del sistema se decide instalar la aplicación *Media Wiki* como un enlace dentro del menú.

El administrador del sistema podrá editar información que considere relevante con esta herramienta con el ánimo de facilitar la configuración a usuarios administradores más inexpertos. De esta forma se permite un mantenimiento en línea de un manual de usuario en el *ESCC*.

La instalación se realiza sobre *XAMPP* ya que esta aplicación contiene todos los elementos necesarios para poder instalar *Mediawiki*.

Los pasos necesarios para su instalación se enumeran a continuación y se parte como requisito inicial tener instalado la aplicación *XAMPP* en el equipo, en este caso el *ESCC*:

Paso 1: Creación de árbol de directorio. Se crea el directorio donde almacenaremos la aplicación:

```
pfc@escc:~$ cd /opt/lampp/htdocs/
pfc@escc:/opt/lampp/htdocs$ sudo mkdir mediawiki
[sudo] password for pfc:
pfc@escc:/opt/lampp/htdocs$ ls
crossdomain.xml  div-dat  favicon.ico  index.php  mediawiki
webalizer  xampp
pfc@escc:/opt/lampp/htdocs$ ls -l
total 56
-rwxr--r--  1 nobody nogroup   173 2008-11-29 19:03 crossdomain.xml
drwxr-xr-x 10 nobody nogroup  4096 2009-10-15 19:07 div-dat
-rw-r--r--  1 root    root     30894 2007-05-11 14:40 favicon.ico
-rwxr--r--  1 root    root     1303 2009-11-01 23:03 index.php
drwxr-xr-x  2 root    root     4096 2009-11-02 15:48 mediawiki
drwxr-xr-x  2 nobody root     4096 2004-12-26 11:30 webalizer
drwxr-xr-x  7 root    root     4096 2008-10-02 15:27 xampp
pfc@escc:/opt/lampp/htdocs$ cd mediawiki/
pfc@escc:/opt/lampp/htdocs/mediawiki$
```

Paso 2: Descarga de la aplicación. Se descarga la aplicación desde la página web oficial del proyecto:

```
pfc@escc:/etc$ sudo wget
http://download.wikimedia.org/mediawiki/1.15/mediawiki-1.15.1.tar.gz
--2009-11-02 15:58:16--
http://download.wikimedia.org/mediawiki/1.15/mediawiki-1.15.1.tar.gz
Resolviendo download.wikimedia.org... 208.80.152.183
Conectando a download.wikimedia.org[208.80.152.183]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 10826029 (10M) [application/octet-stream]
Guardando: «mediawiki-1.15.1.tar.gz»

100%[=====>] 10.826.029 143K/s
en 43s

2009-11-02 15:58:59 (249 KB/s) - `mediawiki-1.15.1.tar.gz' guardado
[10826029/10826029]
```

Paso 3: Descomprimir el archivo en el directorio creado. Se descomprime y desempaqueta la aplicación sobre el directorio creado anteriormente:


```
pfc@escr:/etc$ sudo tar xvfz mediawiki-1.15.1.tar.gz -C
/opt/lampp/htdocs/mediawiki/
```

```
[salida omitida]...
mediawiki-1.15.1/thumb.php
mediawiki-1.15.1/trackback.php5
mediawiki-1.15.1/COPYING
mediawiki-1.15.1/img_auth.php
mediawiki-1.15.1/CREDITS
mediawiki-1.15.1/wiki.phtml
```

Paso 5: Copiamos el contenido de la carpeta al directorio creado. Almacenamos la versión de *MediaWiki* descargada en el directorio raíz de mediawiki para futuras actualizaciones se quedan almacenadas en la misma ubicación:

```
pfc@escr:/opt/lampp/htdocs/mediawiki/mediawiki-1.15.1$ sudo cp *
/opt/lampp/htdocs/mediawiki/ -R
```

Paso 4: Crear la *base de datos*. Se crea la base de datos '*mediawiki*' con la aplicación *phpMyAdmin*.

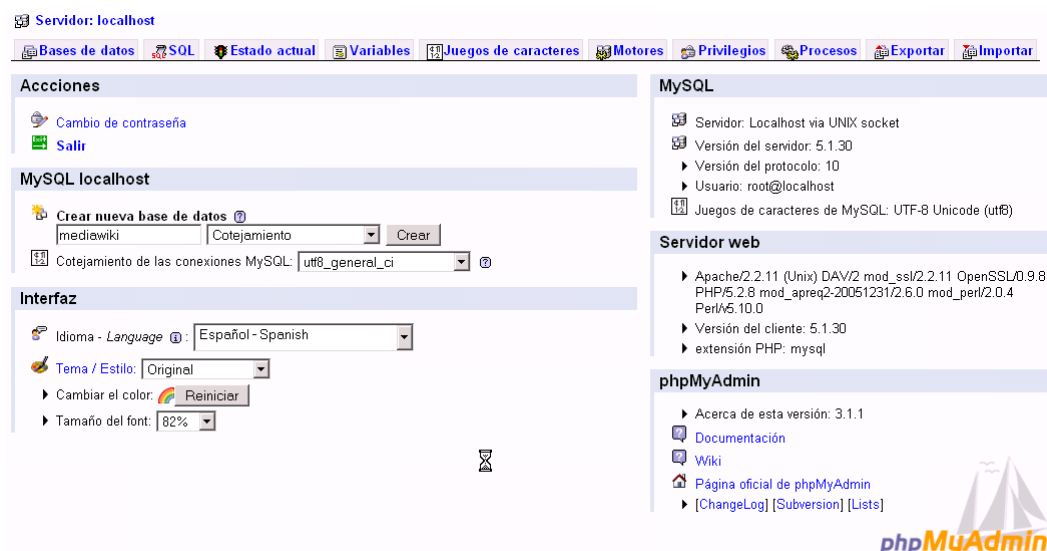


Ilustración 144.-. Creación de la base de datos por aplicación Mediawiki.

Comprobamos que la *base de datos* se ha creado correctamente:



Ilustración 145.-. Comprobación de creación de la base de datos mediante phpAdmin

Paso 6: Instalar MediaWiki. Accedemos a través del navegador a la página de inicio de la aplicación.



Pulsamos en el enlace de configuración de la aplicación y seguimos los pasos que nos indican:

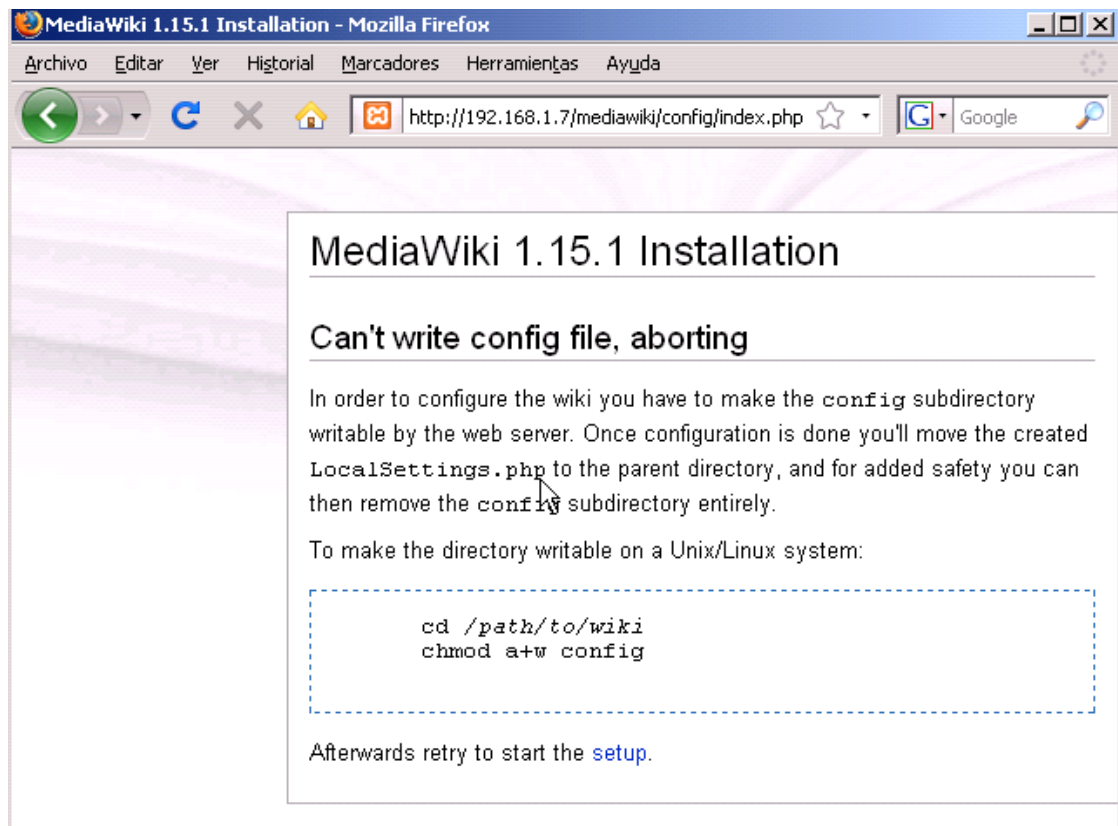


Ilustración 146.- Pantalla de inicio de la instalación de MediaWiki.

Por tanto aplicamos los permisos de escritura a la carpeta config:

```
pfc@escc:/opt/lampp/htdocs/mediawiki$ sudo chmod a+w config
```

Pulsamos sobre el hipervínculo *setup* y aparece un formulario que se rellena teniendo en cuenta el súper usuario a la *base de datos* (root) y la instalamos sobre MySQL.

Comprobamos que la instalación se realiza de forma apropiada:

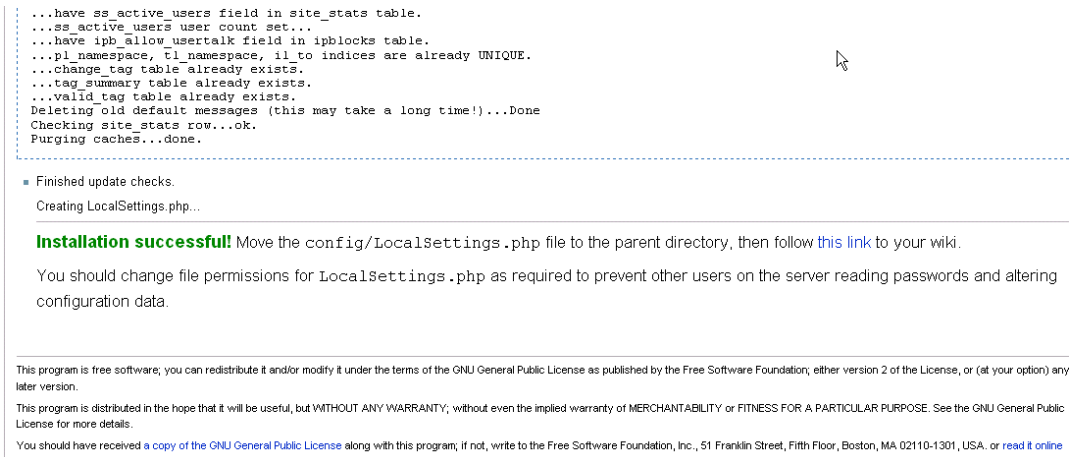


Ilustración 147.- Resultado de la instalación MediaWiki.

Y realizamos las indicaciones que aparecen en la ilustración anterior:

```
pfc@escc:/opt/lampp/htdocs/mediawiki$ sudo mv
config/LocalSettings.php LocalSettings.php
```

Comprobamos que se accede a la página de mediawiki:

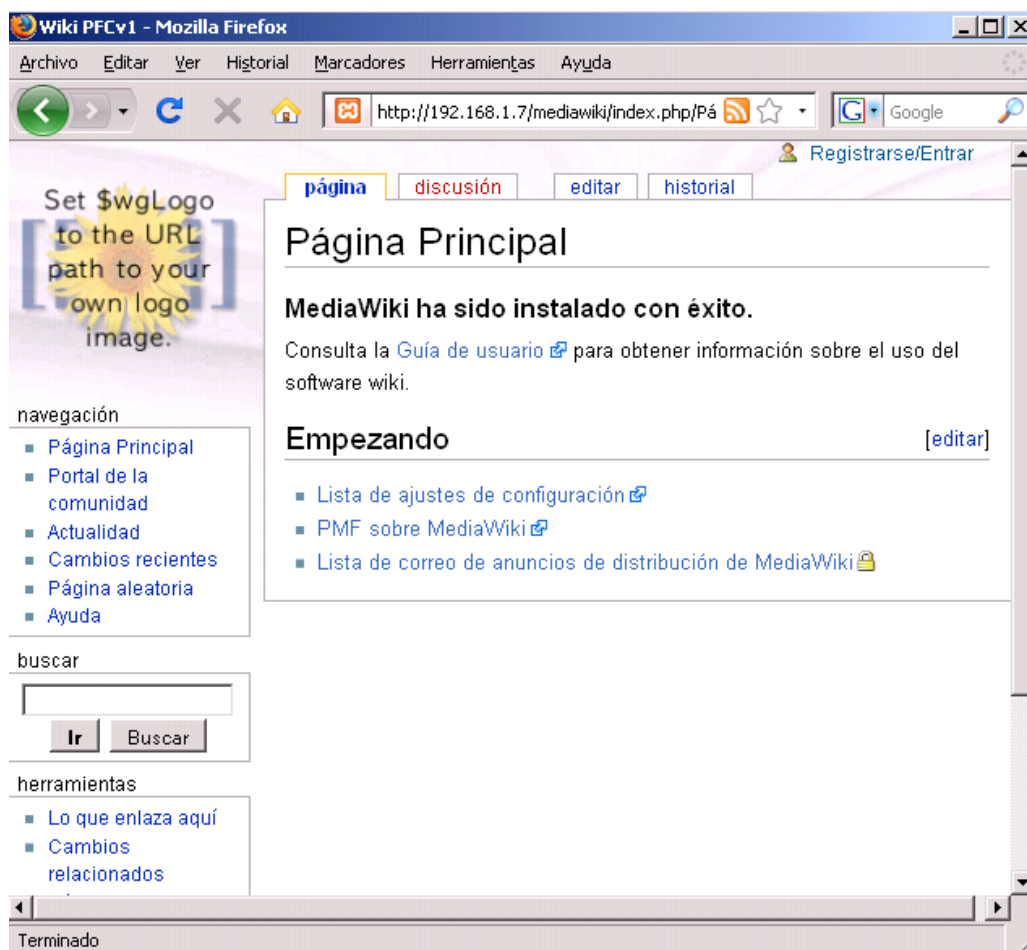


Ilustración 148.- Página principal de la aplicación Media Wiki.

Ya se tiene instalado el sistema *MediaWiki*. El usuario administrador puede ahora agregar contenido al sistema.

E Archivo de Políticas URL para las Aplicaciones SWF

Es necesaria la presencia del *archivo de políticas url* debido a que las *aplicaciones swf*, en concreto los *Reproductores de Zona (repZona.swf)*, se encuentran en el *ESCC* y establecen conexión a otros equipos (los propios ERC) mediante peticiones *http*.

Para establecer la arquitectura *Auto-Proxy* es necesario que a los *Reproductores de Zona* se les permita reproducir archivos de forma local por lo que la aplicación debe recibir un *archivo de políticas url*. El archivo necesariamente se debe denominar *crossdomain.xml* y se almacena en el directorio raíz del servidor *Apache* (aplicación *XAMPP*).

Se ha escrito en el *ESCC* un *archivo de políticas url* totalmente permisivo:

```
<?xml version="1.0"?>
<!-- http://dtimonPFC/crossdomain.xml -->
<cross-domain-policy>
    <allow-access-from domain="*" to-ports="*" />
</cross-domain-policy>
```

BIBLIOGRAFÍA Y REFERENCIAS

1. **Wikipedia Edición Castellano.** Wikipedia. [En línea] [Citado el: 10 de febrero de 2010.] <http://es.wikipedia.org/wiki/Wikipedia>.
2. —. Narrowcasting. [En línea] [Citado el: 20 de diciembre de 2009.] <http://es.wikipedia.org/wiki/Narrowcasting>.
3. **Information Processing Techniques Office (DARPA).** RFC 791. STD5. Internet Protocol. *Defense Advanced Research Projects Agency*. [En línea] septiembre de 1981. <https://datatracker.ietf.org/doc/rfc791/>.
4. **Wikipedia Edición Catellano.** IP Multicast. [En línea] [Citado el: 6 de febrero de 2010.] http://es.wikipedia.org/wiki/IP_Multicast.
5. **Wikipedia Edición Castellano.** Broadcast. [En línea] [Citado el: 6 de febrero de 2010.] http://es.wikipedia.org/wiki/Broadcast_%28inform%C3%A1tica%29.
6. **Rosas, Gabriel Andrés Martínez y Carrasco, Rodrigo Emilio Jaramillo.** IPTV, Análisis de la Tecnología Sobre Distintos Medios de Transmisión e Impacto en el Núcleo de la Red Causados Por Servicios Unicast y Multicast". *Tesis Doctoral. Ingeniería Electrónica. Universidad Austral de Chile*. [En línea] 2008. [Citado el: 19 de abril de 2010.] cybertesis.uach.cl/tesis/uach/2008/bmfcim385i/doc/bmfcim385i.pdf.
7. **Wikipedia Edición Castellano.** Cookies. [En línea] [Citado el: 7 de febrero de 2010.] <http://es.wikipedia.org/wiki/Cookies>.
8. **Huidobro, José Manuel y Roldán Martínez, David,.** *Redes y Servicios de banda ancha*. s.l. : McGraw-Hill. Interamericana de España, S.A, 2004. 84-481-4026-5.
9. **Austerberry, David.** *Tecnología Streaming de Video y Audio*. Andoain : Donostiarra - España, 2005. 8493344575.
10. **Huidobro Moya, José Manuel.** *Manual de Telecomunicaciones*. Paracuellos de Jarama : RA-MA, 2004. 8478975829.
11. **Moya Huidobro, José Manuel y Martínez Roldán, David.** *Integración de voz y datos*. [ed.] Antonio García Brague. Madrid : McGraw-Hill Profesional, 2003. 8448138503.
12. **Departamento de Telemática Universidad Carlos III de Madrid.** Apuntes de Redes y Servicios. Año académico 2003-2004. [En línea] 2004. <http://www.it.uc3m.es/~prometeo/rsc/index.html>.
13. **Cisco.** Cisco Accelerated Internet over Satellite Solution. [En línea] [Citado el: 15 de febrero de 2010.] http://www.cisco.com/en/US/prod/collateral/modules/ps2797/solution_overview_c07-525404.html.

14. **Wikipedia Edición Castellano**. Redes Inlámbricas de Nueva Generación. [En línea] [Citado el: 9 de febrero de 2010.] http://es.wikipedia.org/wiki/Redes_inal%C3%A1mbricas_de_nueva_generaci%C3%B3n.
15. **Barnés, Rodrigo López**. Red Basada en Acceso Inalámbrico (wifi & wimax). *Proyecto Final de Carrera. Universidad Autónoma de Madrid*. [En línea] marzo de 2008. [Citado el: 3 de marzo de 2010.] <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20080409RodrigoLopez.pdf>.
16. **Wikipedia Edición Inglés**. Progressive Download. [En línea] [Citado el: 14 de abril de 2010.] http://en.wikipedia.org/wiki/Progressive_download.
17. **Streamingmedia**. Streaming vs Downloading Video: Choosing de Right Solution. [En línea] 2004. [Citado el: 20 de abril de 2010.] <http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=64573>.
18. **Wikipedia Edición Inglés**. Streaming Media. [En línea] [Citado el: 11 de abril de 2010.] http://en.wikipedia.org/wiki/Streaming_media.
19. **Venegas M., Mauricio, Yáñez C., Aquiles y J. González, Agustín**. Transmisión de Video de Alta Calidad a Través de Redes IP Utilizando Herramientas de Código Abierto. [En línea] 2006. [Citado el: 10 de julio de 2010.] <http://profesores.elo.utfsm.cl/~agv/publications/2006/senacitel/YanezVenegasGonzalezSubmitted.pdf>.
20. **Wikipedia Edición Castellano**. Streaming. [En línea] [Citado el: 12 de noviembre de 2010.] <http://es.wikipedia.org/wiki/Streaming>.
21. **H. Schulzrinne (GMD Fokus)**. RFC 1890. RTP Profile for Audio and Video Conferences with Minimal Control. *IETF. Network Working Group. Audio-Video Transport Working Group*. [En línea] enero de 1996. [Citado el: 25 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc1890/>.
22. **H. Schulzrinne (Columbia University); S. Casner (Packet Design)**. RFC 3551. RTP Profile for Audio and Video Conferences with Minimal Control. *IETF. Network Working Group*. [En línea] julio de 2003. [Citado el: 26 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc3551/>.
23. **C. Perkins (University of Glasgow); M. Westerlund (Ericsson)**. RFC 5761. Multiplexing RTP Data and Control Packets on a Single Port. *Internet Engineering Task Force (IETF)*. [En línea] abril de 2010. [Citado el: 27 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc5761/>.
24. **M. Baugher, D. McGrew (Cisco Systems, Inc.); M. Naslund, E. Carrara, K. Norrman (Ericsson Research)**. RFC 3711. The Secure Real-time Transport Protocol (SRTP). *IETF. Network Working Group*. [En línea] marzo de 2004. [Citado el: 26 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc3711/>.
25. **H. Schulzrinne (Columbia U.); A. Rao (Netscape); R. Lanphier (RealNetworks)**. RFC 2326. Real Time Streaming Protocol (RTSP). *IETF. Network Working Group*. [En línea] abril de 1998. [Citado el: 24 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc2326/>.

26. **Active Internet-Draft (MMUSIC WG document).**MMUSIC Working Group. Real Time Streaming Protocol 2.0 (RTSP). [En línea] 17 de noviembre de 2010. <https://datatracker.ietf.org/doc/draft-ietf-mmusic-rfc2326bis/>.
27. **Wikipedia Edición Castellano.** HTTP: Hypertext Transfer Protocol. [En línea] [Citado el: 20 de marzo de 2010.] <http://es.wikipedia.org/wiki/Http>.
28. **R. Fielding (UC Irvine);J. Gettys (Compaq/W3C);J. Mogul (Compaq);H. Frystyk (W3C/MIT); L. Masinter(Xerox);P. Leach (Microsoft);T. Berners-Lee(W3C/MIT).** RFC 2616. Hypertext Transfer Protocol HTTP/1.1. *IETF. Network Working Group.* [En línea] junio de 1999. [Citado el: 1 de febrero de 2010.] <https://datatracker.ietf.org/doc/rfc2616/>.
29. **Adobe.** Real Time Messaging Protocol Chunk Stream. [En línea] [Citado el: 7 de mayo de 2010.] http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf.
30. **Osflash:RTMP.** RTMP Protocol [DRAFT]. [En línea] [Citado el: 9 de junio de 2010.] <http://osflash.org/documentation/rtmp>.
31. **Adobe.** Flash Media Server para la Transmisión de Video. [En línea] [Citado el: 15 de mayo de 2010.] <http://www.adobe.com/es/products/flashmediaserver/>.
32. **Realnetworks: Stream Video.** Stream Video to PC and Mobile Phones. [En línea] [Citado el: 15 de mayo de 2011.] <http://www.realnetworks.com/products-services/helix-server-proxy.aspx>.
33. **WowzaMedia: Media Servers.** Flash, Iphone, Sylverlight Media Server and More. [En línea] [Citado el: 15 de mayo de 2011.] <http://www.wowzamedia.com/>.
34. **Red5:Documentation.** Documentation Red5. [En línea] [Citado el: 15 de mayo de 2010.] <http://trac.red5.org/wiki/Documentation>.
35. **Wikipedia Edición Castellano.** Codec. [En línea] [Citado el: 4 de enero de 2011.] <http://es.wikipedia.org/wiki/Codec>.
36. —. Audio Video Interleave. [En línea] [Citado el: 7 de mayo de 2010.] <http://es.wikipedia.org/wiki/AVI>.
37. **Microsoft.** AVI RIFF File Reference. [En línea] [Citado el: 5 de enero de 2011.] <http://msdn.microsoft.com/en-us/library/ms779636%28v=vs.85%29.aspx>.
38. **Wikipedia Edición Castellano.** Ogg. [En línea] [Citado el: 4 de mayo de 2010.] <http://es.wikipedia.org/wiki/Ogg>.
39. **S. Pfeiffer (CSIRO).** RFC 3533. The Ogg Encapsulation Format Version 0. *ITEF. Network Working Group.* [En línea] mayo de 2003. [Citado el: 5 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc3533/>.

40. **L. Walleij (The Ogg Vorbis Community)**. RFC 3534. The application/ogg Media Type. *IETF Working Group*. [En línea] mayo de 2003. [Citado el: 5 de mayo de 2010.] <https://datatracker.ietf.org/doc/rfc3534/>.
41. **Fundación xiph.org/**. Documentación Ogg. [En línea] 1994. [Citado el: 5 de mayo de 2010.] <http://xiph.org/ogg/doc/>.
42. **Wikipedia Edición Castellano**. Ogg Media. [En línea] [Citado el: 5 de mayo de 2010.] http://es.wikipedia.org/wiki/Ogg_Media.
43. —. Advanced Streaming Format. [En línea] [Citado el: 5 de mayo de 2010.] http://es.wikipedia.org/wiki/Advanced_Streaming_Format.
44. —. MPEG-4 Parte 14. [En línea] [Citado el: 4 de mayo de 2010.] http://es.wikipedia.org/wiki/MPEG-4_Parte_14.
45. —. Matroska. [En línea] [Citado el: 5 de mayo de 2010.] <http://es.wikipedia.org/wiki/Matroska>.
46. —. Flash Video. [En línea] [Citado el: 4 de mayo de 2010.] <http://es.wikipedia.org/wiki/Flv>.
47. **Adobe**. Guía de Aprendizaje de Flash video. [En línea] [Citado el: 4 de mayo de 2010.] http://www.adobe.com/es/devnet/flash/articles/video_guide_print.html.
48. **Gea, Juan Pedro Muñoz, y otros**. Evaluación de la Tecnología HomePlug AV para la Provisión de Servicios Multimedia en el Hogar. *Departamento de Tecnologías de la Información y las Comunicaciones. Universidad Politécnica de Cartagena*. [En línea] 2008. [Citado el: 19 de abril de 2010.] repositorio.bib.upct.es/dspace/bitstream/10317/1365/1/eth.pdf.
49. **Microsoft**. Comparing HTTP Streaming Protocol with RTSP. [En línea] [Citado el: 15 de mayo de 2010.] <http://msdn.microsoft.com/en-us/library/bb905764.aspx>.
50. **W. Fenner (Xerox PARC)**. RFC 2236. Internet Group Management Protocol, Version 2. *IETF Network Working Group*. [En línea] noviembre de 1997. [Citado el: 21 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc2236/>.
51. **B. Cain (Cereva Networks); S. Deering, I. Kouvelas (Cisco Systems); B. Fenner (AT&T Labs - Research); A. Thyagarajan (Ericsson)**. RFC 3376. Internet Group Management Protocol, Version 3. *IETF Network Working Group*. [En línea] octubre de 2002. [Citado el: 20 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc3376/>.
52. **VideoLAN**. Aplicación VideoLAN Client. [En línea] [Citado el: 14 de septiembre de 2010.] <http://www.videolan.org>.
53. **VNC**. Aplicación VNC Server. *Aplicación VNC Server*. [En línea] 2009. <http://www.vnc.com>.
54. **VideoLAN**. The HTTP interface. *Wiki VideoLAN*. [En línea] [Citado el: 27 de diciembre de 2010.] <http://www.videolan.org/doc/play-howto/en/ch04.html#id590873>.

55. **VMWARE.** Aplicación VMWARE Server. [En línea] [Citado el: 24 de enero de 2010.] http://www.vmware.com/files/es/pdf/09Q1_VM_SERVER_DS_ES_A4_R1.pdf.
56. **Emule.** Aplicación Emule. *Aplicación Emule*. [En línea] 2007. <http://www.emule-project.net>.
57. **Amule.** Aplicación amule para sistemas Linux. *Aplicación amule para sistemas Linux*. [En línea] 2009. <http://www.amule.org>.
58. **Cisco.** Cisco Digital Media System: Comprehensive. Scalable. Network-Centric. *Cisco Product Brochure*. [En línea] 2008. <http://www.cisco.com>.
59. —. Cisco Digital Signage: Flexible and Centralized Management of Digital Signage Displays. *Cisco Product Brochure*. [En línea] 2008. http://www.cisco.com/en/US/prod/collateral/video/ps9339/ps7220/prod_brochure0900aecd805ae843.pdf.
60. —. Wide Area Application Services. [En línea] [Citado el: 6 de marzo de 2010.] www.cisco.com/go/waas.
61. —. Configuring Cisco Access Routers and the NME-WAE Network Module for ACNS Deployments. [En línea] [Citado el: 4 de marzo de 2010.] http://www.cisco.com/en/US/docs/app_ntwk_services/waas/acns/v55/module/configuration/guide/acnsnme.html.
62. —. Cisco ACNS Software Configuration Guide for Centrally Managed Deployments. [En línea] [Citado el: 9 de enero de 2011.] http://www.cisco.com/en/US/docs/app_ntwk_services/waas/acns/v55/configuration/central/guide/55cdg.html.
63. **Servidor Darwing.** Darwing Streaming Server: DSS. [En línea] [http://dss.macosforge.org/..](http://dss.macosforge.org/)
64. **Wikipedia Edición Castellano.** Rich Internet Applications. [En línea] [Citado el: 14 de febrero de 2010.] http://es.wikipedia.org/wiki/Rich_Internet_Application.
65. **Cisco.** Cisco Digital Media System Deployments for Cisco Desktop Video with Cisco ACNS. *Cisco White Paper*. [En línea] 2008. <http://www.cisco.com>.
66. —. DMS Design Guide. [En línea] [Citado el: 4 de enero de 2011.] http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/DMS_DG/DMS_dg.html.
67. **Comunicads Dynamic Advertising.** Arquitectura de Producto Comunicads (último acceso: 01 de diciembre de 2010). [En línea] [Citado el: 1 de diciembre de 2010.] <http://comunicads.com/arquitectura.html>.
68. **IDC.** Retail IT Meets Video: Cisco Makes Digital Signage Play. *Sponsored by Cisco*. [En línea] 2007. [Citado el: 2 de diciembre de 2010.] http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns158/ns620/net_implementation_white_paper0900aecd806b8c04.pdf.

69. **Fractalia Systems.** Fractalia Media Whitepaper (último acceso: 7 de diciembre de 2010). [En línea] [Citado el: 7 de diciembre de 2010.] http://www.fractaliasystems.com/file/fractalia_media_whitepaper_es_4_0_0.pdf.
70. **Wikipedia Edición Castellano.** Peer-to-Peer. [En línea] [Citado el: 20 de enero de 2011.] <http://es.wikipedia.org/wiki/Peer-to-peer>.
71. **Millán Tejedor, Ramon Jesús.** Guía Fácil de las TIC. [En línea] 2006. [Citado el: 3 de diciembre de 2010.] <http://www.ramonmillan.com/tutoriales/guaticp2p.php>.
72. **p2pfoundation.** The Foundation for P2P Alternatives. [En línea] [Citado el: 1 de diciembre de 2010.] <http://p2pfoundation.net/>.
73. **ISO/IEC.** *Information Technology - Open systems - Basic Reference Model: The Basic Model.* Ginebra, Suiza : ISO/IEC, 1994. ISO/IEC 7498-1.
74. **OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. Zimmermann, Hubert.** s.l. : IEEE Transactions on Communications, 1980. Vol.COM-28, NO.4, Abril 1980.
75. **Adobe.** Plataforma Adobe Flash. [En línea] [Citado el: 1 de marzo de 2010.] <http://www.adobe.com/es/flashplatform/>.
76. —. Sitio Oficial de AIR de Adobe. [En línea] [Citado el: 20 de marzo de 2010.] <http://www.adobe.com/es/products/air/>.
77. **ECMA International. EcmaScript.org.** Overview.Especificaciones de ECMA Edición 4. [En línea] 2007. [Citado el: 21 de marzo de 2010.] <http://www.ecmascript.org/es4/spec/overview.pdf>.
78. —. Compatibility Between ES3 and Proposed ES4. [En línea] noviembre de 2007. [Citado el: 21 de marzo de 2010.] <http://www.ecmascript.org/es4/spec/incompatibilities.pdf>.
79. —. ECMAScript 4th Edition Grammar. [En línea] 16 de mayo de 2008. [Citado el: 30 de marzo de 2010.] <http://www.ecmascript.org/es4/spec/grammar-2008-05-16.pdf>.
80. **Wikipedia Edición Inglés.** ActionScript. [En línea] [Citado el: 4 de marzo de 2010.] <http://en.wikipedia.org/wiki/ActionScript>.
81. **ECMA International.** Especificaciones de ECMAScript for XML (E4X). [En línea] <http://www.ecma-international.org/publications/standards/Ecma-357.htm>.
82. **Adobe.** Aprendizaje de ActionScript 3.0. *Adobe Help.* [En línea] http://help.adobe.com/es_ES/as3/learn/as3_learning.pdf.
83. **Microsoft:Silverlight.** Silverlight. [En línea] [Citado el: 10 de marzo de 2010.] <http://www.microsoft.com/silverlight/>.
84. **Wikipedia Edición Castellano.** Adobe Flash Player. [En línea] [Citado el: 14 de abril de 2010.] http://es.wikipedia.org/wiki/Adobe_Flash_Player.

85. **Oracle.** JavaFX. [En línea] [Citado el: 16 de marzo de 2010.] <http://www.oracle.com/us/products/tools/050854.html>.
86. **Deitel, Paul J. y Deitel, Harvey M.** *Ajax, Rich Internet Applications y Desarrollo Web para Programadores*. Madrid : Anaya Multimedia, 2008. ISBN: 978-84-415-2467-5.
87. **Wikipedia Edición Castellano.** Ajax. [En línea] [Citado el: 9 de marzo de 2010.] <http://es.wikipedia.org/wiki/AJAX>.
88. **ASP (Active Server Page).** AJAX: The Official Microsoft ASP.NET AJAX Site. *Ajax: Enhanced Interactivity and Responsiveness*. [En línea] [Citado el: 10 de marzo de 2010.] <http://www.asp.net/ajax>.
89. **xajax-project.org.** Xajax. Ajax for PHP. [En línea] [Citado el: 7 de marzo de 2010.] <http://xajax-project.org/en/home/>.
90. **Glass, Michael, y otros.** *Desarrollo web con PHP, apache y MySQL*. Madrid : Anaya Multimedia, 2004. ISBN:84-415-1755-X.
91. **Apache Friends.** Página Oficial del Proyecto XAMPP. [En línea] <http://www.apachefrineds.org>.
92. **IBM.** Características de XAMPP. *Características de XAMPP*. [En línea] 2007. [Citado el: 26 de marzo de 2010.] <http://www.ibm.com/developerworks/linux/library/l-xampp/?ca=dgr-wikiaXAMPP>.
93. **Reinhardt, Robert y Dowd, Snow.** *Adobe Flash CS3 (La Biblia)*. Madrid : Anaya Multimedia, 2008. ISBN:9788441523937.
94. **Siever, Ellen, y otros.** *La guía definitiva sobre Linux*. Madrid : Anaya Multimedia, 2010. ISBN:978-84-415-2725-6.
95. **Ramos, Antonio Ángel, y otros.** *Instala, administra, securiza y virtualiza Entornos Linux*. Madrid : Ra-Ma Editorial, 2008. ISBN:978-84-7897-896-0.
96. **Wikipedia Edición Castellano.** Máquina Virtual. [En línea] [Citado el: 17 de febrero de 2010.] http://es.wikipedia.org/wiki/M%C3%A1quina_virtual.
97. **VMware.** Aspectos Básicos de la Virtualización. [En línea] [Citado el: 5 de febrero de 2010.] <http://www.vmware.com/es/virtualization/virtual-machine.html>.
98. **Wikipedia Edición Castellano.** Control de Versiones. [En línea] [Citado el: 8 de enero de 2011.] http://es.wikipedia.org/wiki/Control_de_versiones.
99. **MySQL:Manual.** How MySQL Uses DNS. [En línea] [Citado el: 12 de septiembre de 2010.] <http://dev.mysql.com/doc/refman/5.5/en/dns.html>.
100. **Nemeth, Evi, y otros.** *Unix and Linux Systems Administration Hnadbook. Fourth Edition*. Boston : Prentice Hall, 2010. ISBN: 978-0-13-148005-6.

101. **Oficina Nacional de Tecnologías de Información (ONTI)**. . Instalación y configuración de NTP. *ArCERT. Coordinación de Emergencias en Redes Teleinformáticas de Argentina*. [En línea] 2006. [Citado el: 7 de mayo de 2010.] <http://www.arcert.gov.ar/webs/tips/NTPv1.0.pdf>.
102. **Wikipedia Edición Castellano**. NTP. [En línea] [Citado el: 7 de abril de 2010.] <http://es.wikipedia.org/wiki/NTP>.
103. **Meketa, Deneb**. Policy file changes in Flash Player 9 and Flash Player 10 . [En línea] 2008. [Citado el: 24 de diciembre de 2010.] http://www.adobe.com/devnet/flashplayer/articles/fplayer9_security.html.
104. **Adobe**. Controles de Sitio Web (archivos de política). *ActionScript 3. Adobe Help*. [En línea] [Citado el: 29 de marzo de 2010.] http://help.adobe.com/es_ES/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7e08.html.
105. **Uhley, Peleus**. Setting up a socket policy file server. [En línea] April de 2008. [Citado el: 15 de diciembre de 2009.] http://www.adobe.com/devnet/flashplayer/articles/socket_policy_files.html.
106. **Adobe**. Understanding Flash Player 9 April 2008 Security Update compatibility. [En línea] 2008. [Citado el: 20 de diciembre de 2009.] http://www.adobe.com/devnet/flashplayer/articles/flash_player9_security_update.html.
107. **IANA**. Port Number. [En línea] [Citado el: 24 de febero de 2011.] <http://www.iana.org/assignments/port-numbers>.
108. **Wikipedia Edición Inglés**. List of TCP and UDP Port Numbers. [En línea] [Citado el: 15 de marzo de 2010.] http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.
109. **Lauritzen, Ammon**. Flash Policy Service Daemon. [En línea] 2008. [Citado el: 5 de enero de 2010.] <http://ammonlauritzen.com/blog/2008/04/22/flash-policy-service-daemon/>.
110. **Heinaaro, Kimmo**. Addon R-kiosk. [En línea] 2007. [Citado el: 07 de noviembre de 2010.] <https://addons.mozilla.org/es-es/firefox/addon/r-kiosk/>.
111. **VideoLAN**. Documentation:WebPlugin. *Wikimedia VideoLAN*. [En línea] [Citado el: 7 de diciembre de 2010.] <http://wiki.videolan.org/Documentation:WebPlugin>.
112. **S. Deering (Stanford University)**. RFC 1112. Host Extensions for IP Multicasting. *IETF. Network Working Group*. [En línea] agosto de 1989. [Citado el: 21 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc1112/>.
113. **B. Fenner (AT&T Labs - Research); M. Handley (UCL); H. Holbrook (Arastra); I. Kouvelas(Cisco)**. RFC 4601. Protocol Independent Multicast - Sparse Mode (PIM-SM):Protocol Specification (Revised). *IETF.Network Working Group*. [En línea] agosto de 2006. [Citado el: 10 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc4601>.
114. **A. Adams (NextHop Technologies); J. Nicholas (ITT A/CD); W. Siadak (NextHop Technologies)**. RFC 3973. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol

Specification (Revised). *IETF. Network Working Group*. [En línea] enero de 2005. [Citado el: 10 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc3973>.

115. **VideoLAN**. Documentation:Play HowTo/Advanced Use of VLC. *Wikimedia VideoLAN*. [En línea] [Citado el: 25 de enero de 2011.] http://wiki.videolan.org/Documentation:Play_HowTo/Advanced_Use_of_VLC#Install_the_plugin.

116. **World Wide Web Consortium**. Synchronized Multimedia. [En línea] 2008. [Citado el: 14 de enero de 2011.] <http://www.w3.org/AudioVideo/>.

117. —. Synchronized Multimedia Integration Language (SMIL 3.0). [En línea] 1 de diciembre de 2008. [Citado el: 14 de enero de 2011.] <http://www.w3.org/TR/2008/REC-SMIL3-20081201/>.

118. **Ambulant Player**. Ambulant Open SMIL Player. [En línea] [Citado el: 14 de febrero de 2011.] <http://www.ambulantplayer.org/>.

119. **Apple**. ¿Qué es QuickTime 7? [En línea] [Citado el: 7 de febrero de 2011.] <http://www.apple.com/es/quicktime/what-is/>.

120. **Mozilla**. Plugins:Add-ons for Firefox. [En línea] [Citado el: 7 de febrero de 2011.] <https://addons.mozilla.org/en-US/firefox/browse/type:7>.

121. **Apple**. SMIL Scripting Guide for QuickTime. [En línea] 5 de abril de 2005. [Citado el: 19 de septiembre de 2010.] http://developer.apple.com/legacy/mac/library/documentation/QuickTime/Conceptual/QTScripting_SMIL/QTScripting_SMIL.pdf.

122. —. JavaScript Scripting Guide for QuickTime. [En línea] [Citado el: 7 de febrero de 2011.] http://developer.apple.com/library/safari/#documentation/QuickTime/Conceptual/QTScripting_JavaScript/aQTScripting_Javascro_AIntro/Introduction%20to%20JavaScript%20QT.html#//apple_ref/doc/uid/TP40001526-IntroCH001-DontLinkElementID_21.

123. **Gecko Media Player**. Gecko Media Player:gecko-mediaplayer . [En línea] [Citado el: 7 de febrero de 2011.] <http://code.google.com/p/gecko-mediaplayer/>.

124. **Mozilla**. Usar Silverlight en Firefox. [En línea] [Citado el: 7 de febrero de 2011.] <http://support.mozilla.com/es/kb/Usar%20Silverlight%20en%20Firefox>.

125. **Mono-Project**. Moonlight. [En línea] septiembre de 2007. [Citado el: 7 de febrero de 2011.] <http://www.mono-project.com/Moonlight>.

126. **Microsoft: MoonLight**. Covenant to Downstream Recipients of Moonlight - Microsoft & Novell Interoperability Collaboration. [En línea] [Citado el: 7 de febrero de 2011.] <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/IPLicensing/customer covenant/msnovellcollab/moonlight.aspx>.

127. **LIRC**. Linux Infrared Remote Control. [En línea] [Citado el: 29 de diciembre de 2010.] <http://www.lirc.org/>.

128. **Acosta, Elisa y Ripoll, Ismael.** Receptor de mando a distancia con RT-Linux. [En línea] 1998. [Citado el: 29 de diciembre de 2010.] <http://www.gii.upv.es/personal/iripoll/str/documentos/rtnlinux/www.linuxfocus.org/Castellano/articles/article57.html>.
129. **S. Alexander (Silicon Graphics, Inc.); R. Droms (Bucknell University).** RFC 2132. DHCP Options and BOOTP Vendor Extensions. *IETF. Network Working Group*. [En línea] marzo de 1997. [Citado el: 28 de diciembre de 2010.] <https://datatracker.ietf.org/doc/rfc2132/>.
130. **Wireshark.** Aplicación Wireshark. [En línea] [Citado el: 1 de septiembre de 2010.] <http://www.wireshark.org/>.
131. **MedialInfo.** Aplicación Mediainfo. [En línea] [Citado el: 19 de octubre de 2010.] <http://mediainfo.sourceforge.net/es>.
132. **Mozillazine.org.** About:config entries. [En línea] [Citado el: 12 de abril de 2010.] http://kb.mozillazine.org/About:config_entries.
133. **Mozillazine.** Browser.cache.memory.capacity. [En línea] [Citado el: 17 de agosto de 2010.] <http://kb.mozillazine.org/Browser.cache.memory.capacity>.
134. **Cisco.com.** DMS Design Guide. [En línea] [Citado el: 4 de enero de 2011.] http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/DMS_DG/DMS_dg.html.
135. **Wireshark.** Aplicación Wireshark. [En línea] [Citado el: 1 de septiembre de 2010.] <http://www.wireshark.org/>.
136. **Ubuntu.** Sistema Operativo Ubuntu 10.10. [En línea] [Citado el: 7 de marzo de 2010.] <http://www.ubuntu.com/>.
137. **Adobe.** Conexiones Socket en ActionScript 3. *Adobe Help*. [En línea] 2008. [Citado el: 15 de abril de 2010.] http://help.adobe.com/es_ES/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7c60.html.
138. **David L. Mills (University of Delaware).** RFC 1305. Network Time Protocol (Version 3). *IETF. Network Working Group*. [En línea] marzo de 1992. [Citado el: 7 de abril de 2010.] <https://datatracker.ietf.org/doc/rfc1305/>.
139. **VMWare.** Understanding Full Virtualization, Paravirtualization and Hardware Assist. [En línea] [Citado el: 24 de enero de 2010.] http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf.
140. **Wikipedia Edición Castellano.** Arduino. [En línea] [Citado el: 12 de diciembre de 2010.] <http://es.wikipedia.org/wiki/Arduino>.
141. **Arduino Project.** Arduino Ethernet Shield. [En línea] 2006. [Citado el: 12 de julio de 2010.] <http://www.arduino.cc/en/Main/ArduinoEthernetShield>.

142. **Noble, Joshua.** *Programming Interactivity. A Designer's Guide to Processing, Arduino, and openFrameworks.* Gravenstein Highway North. Sebastopol. : O'Reilly Media, Inc., 2009. ISBN: 978-0-596-15414-1.
143. **Sparkfun.** Arduino Ethernet Shield. [En línea] enero de 2011. [Citado el: 2 de enero de 2011.] <http://www.sparkfun.com/products/9026>.
144. **ffmpeg.** FFmpeg Project. [En línea] [Citado el: 6 de mayo de 2010.] <http://www.ffmpeg.org/ffmpeg.html>.
145. **Cisco.** Cisco Unified Communications Operating System Administration Guide, Release 8.0(1). [En línea] 2010. [Citado el: 17 de enero de 2011.] http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/8_0_1/cucos/osg_801_cm.pdf.
146. **R. Droms (Bucknell University).** RFC 2939. Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types. *IETF. Network Working Group.* [En línea] septiembre de 2000. [Citado el: 9 de enero de 2011.] <https://datatracker.ietf.org/doc/rfc2939/>.
147. **as3crypto.** Cryptography library for ActionScript 3, including partial TLS 1.0 support. [En línea] [Citado el: 12 de junio de 2010.] <http://code.google.com/p/as3crypto/>.
148. **UCK.** Ubuntu Customization Kit. [En línea] [Citado el: 3 de diciembre de 2010.] <http://uck.sourceforge.net/>.
149. **Cisco.** Configuring InterVLAN Routing and ISL/802.1Q Trunking on a Catalyst 2900XL/3500XL/2950 Switch Using an External Router. [En línea] 22 de noviembre de 2005. [Citado el: 7 de octubre de 2010.] http://www.cisco.com/en/US/tech/tk389/tk815/technologies_configuration_example09186a00800949fd.shtml.
150. **Wikipedia Edición Inglesa.** IEEE 802.1Q. [En línea] [Citado el: 15 de mayo de 2010.] http://en.wikipedia.org/wiki/IEEE_802.1Q. ISBN:0-7381-4877-6.
151. **Wikipedia Edición Castellano.** XAMPP. [En línea] [Citado el: 7 de febrero de 2010.] <http://es.wikipedia.org/wiki/XAMPP>.
152. **Drake, Daniel.** Writing udev rules. [En línea] 2006. [Citado el: 30 de abril de 2010.] http://www.reactivated.net/writing_udev_rules.html.
153. **Collins-Sussman, Ben, W. Fitzpatrick, Brian y Pilato, C. Michael.** *Version Control with Subversion For Subversion 1.6.* Stanford, California : Creative Commons Attribution License, 2009.
154. **Ubuntu.** Sistema Operativo Ubuntu 10.10. [En línea] 2010. [Citado el: 4 de febrero de 2010.] <http://ftp.heanet.ie/pub/ubuntu-cdimage/releases/>.

155. **Lauritzen, Ammon.** Servicios para Políticas Flash. *Servicios para Políticas Flash*. [En línea] 22 de Abril de 2008. [http://ammonlauritzen.com/blog/2008/04/22/flash-policy-service-daemon/..](http://ammonlauritzen.com/blog/2008/04/22/flash-policy-service-daemon/)
156. **Subversion Corporation.** Página Oficial de Subversion: SVN. *Página Oficial de Subversion: SVN*. [En línea] 2010. <http://subversion.apache.org/>.
157. **Adobe.** Origen de las Especificaciones de Action Script 3. *AdobeLivedocs*. [En línea] 2008. http://livedocs.adobe.com/specs/actionscript/3/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=as3_specification.html.
158. **Wikipedia Edición Castellano.** Modelo Vista Controlador. [En línea] [Citado el: 19 de enero de 2010.] http://es.wikipedia.org/wiki/Modelo_vista_controlador.
159. **Anderson, Chris.** *La Economía Long Tail*. Barcelona : Empresa Activa, 2009. ISBN: 9788492452316.
160. **Lobato Pérez, Montserrat.** Introducción a las Máquinas Virtuales. *Introducción a las Máquinas Virtuales*. [En línea] 2008. <http://observatorio.cnice.mec.es>.
161. **Cerebro en la Sombra.** Flash Player, Sockets y Políticas de Seguridad. *Flash Player, Sockets y Políticas de Seguridad*. [En línea] 2008. [http://blog.osusnet.com/2008/04/15/flash-player-sockets-y-politicas-de-seguridad/..](http://blog.osusnet.com/2008/04/15/flash-player-sockets-y-politicas-de-seguridad/)
162. **ECMA International.** Especificaciones de ECMAScript 3 y 5. *Especificaciones de ECMAScript 3 y 5*. [En línea] 2010. <http://wiki.ecmascript.org/doku.php>.
163. **Adobe.** Especificaciones de ActionScript3. *AdobeLivedocs*. [En línea] Adobe, 2006. <http://livedocs.adobe.com/specs/actionscript/3/wwhelp/wwhimpl/js/html/wwhelp.htm>.
164. **ECMA International.** ECMAScript for XML (E4X) Specification. [En línea] diciembre de 2005. [Citado el: 30 de marzo de 2010.] <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-357.pdf>.
165. **Wikipedia Edición Inglés.** ECMAScript. [En línea] 2010. <http://en.wikipedia.org/wiki/ECMAScript>.
166. **Emule.** Aplicación Emule para sistemas Microsoft Windows. *Aplicación Emule para sistemas Microsoft Windows*. [En línea] 2009. <http://www.emule-project.net>.
167. **Moock, Collin.** *Action Script 3*. s.l. : Anaya Multimedia. O`Reilly, 2008. ISBN 978-84-415-2340-1.
168. **Cisco.** Accelerating Video Using Cisco Wide Area Application Services and Digital Media Systems. *Cisco White Paper*. [En línea] 2008. [Citado el: 15 de febero de 2010.] http://www.cisco.com/en/US/prod/collateral/contnetw/ps5680/ps6870/white_paper_c11-499857.pdf.

169. **Junazua.** Free and Open Source P2P File Sharing Software. [En línea] 2008. [Citado el: 12 de diciembre de 2010.] <http://www.junauza.com/2008/01/freeopen-source-p2p-file-sharing.html>.

GLOSARIO

ACNS	<i>Application and Content Networking System (Sistema software de Aplicaciones y Contenido de Red)</i> . Conjunto de <i>software</i> , propiedad de Cisco, que pretende optimizar la distribución de contenido en redes con limitaciones de <i>ancho de banda</i> , con la intención de reducir la congestión de la red almacenando y distribuyendo el contenido en los extremos de la red.
ActionScript	Lenguaje de programación orientado a objetos utilizado, en especial, en aplicaciones web animadas realizadas en el entorno Adobe Flash.
Adobe Air	Es un entorno de ejecución multiplataforma para la construcción de aplicaciones RIA (Rich Internet Applications) utilizando <i>Adobe Flash</i> , <i>Adobe Flex</i> , <i>HTML</i> y <i>AJAX</i> , las cuales pueden usarse como aplicaciones de escritorio.
Adobe Flash	Tecnología de Adobe para añadir dinamismo a las aplicaciones web. Se trata esencialmente de una aplicación de creación y manipulación vectorial y de manejo de código (véase <i>ActionScript</i>) en forma de estudio de animación que trabaja sobre fotogramas y está destinado a la producción y entrega de contenido interactivo para las diferentes audiencias alrededor del mundo sin importar la plataforma.
API	<i>Application Programming Interface (Interfaz de Programación de Aplicaciones)</i> . Es el conjunto de funciones y procedimientos (o métodos) que ofrece cierta biblioteca de programación para ser utilizado por otro <i>software</i> como una capa de abstracción.
ASF	<i>Advanced Streaming Format (Formato de Streaming Avanzado) o Advanced Systems Format (Formato de Sistemas Avanzados)</i> . Formato, propiedad de Microsoft, de archivo contenedor multimedia diseñado especialmente para el <i>Streaming</i> .
ATM	<i>Asynchronous Transfer Mode (Modo de Transferencia Asíncrona)</i> . Tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones que permite transferencia de modo asíncrono; es decir, permite el envío diferentes flujos de información, de características distintas en cuanto a velocidad y formato.
Auto-Proxy	Es el un mecanismo definido en el proyecto fin de carrera en el que el propio <i>ERC</i> (véase <i>ERC</i>) hace las veces de proxy del <i>ESCC</i> (véase <i>ESCC</i>), para recuperar el contenido multimedia.
AVI	<i>Audio Video Interleave (Intercalado de Video y Audio)</i> . Es un formato contenedor multimedia lanzado por Microsoft en 1992 (véase <i>Interleave</i>).
Bit Rate	<i>Tasa de Bits</i> . Es el número de bits que se transmiten por unidad de tiempo a través de un sistema.
Bluetooth	Especificación industrial para <i>Redes Inalámbricas de Área Personal</i> (véase <i>WPAN</i>) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Buffer	Es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras que está esperando para ser procesada.
Caché	Es un conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en tiempo, respecto a la copia en la caché. En ocasiones cuando se accede por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso medio al dato sea menor.
CIR	<i>Committed Information Rate (Tasa de Información Comprometida)</i> . Es el caudal o tasa mínima de información que garantiza el operador o proveedor de servicios de internet (véase <i>ISP</i>) al cliente o usuario. El resto del ancho de banda disponible para el usuario está sujeto al estado de la red y las necesidades del operador.
Códec	Es una especificación que utiliza un dispositivo o programa para desempeñar transformaciones bidireccionales sobre datos y señales. Existen codecs de transmisión, compresión y encriptación. El concepto proviene de <u>codificación</u> y <u>decodificación</u> .
Cracker	Es cualquier persona que viola la seguridad de un sistema informático de forma similar a como lo haría un hacker, sólo que a diferencia de este último, el cracker realiza la intrusión con fines de beneficio personal o para hacer daño. El término deriva de la expresión 'criminal hacker'.
CSS	<i>Cascading Style Sheets (Hoja de Estilo en Cascada)</i> . Es un lenguaje usado para definir la presentación de un documento estructurado escrito en <i>HTML</i> , <i>XML</i> y <i>XHTML</i> . El <i>W3C</i> es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.
DHCP	<i>Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host)</i> .
DMP	<i>Digital Media Player (Reproductor de Medios Digitales)</i> . Es el nombre que ha proporcionado la solución de referencia de <i>Cisco</i> a los equipos terminales que proporcionan la reproducción del contenido multimedia a un sistema de presentación de contenido.
DMM	<i>Digital Media Manager</i> . Aplicación de <i>Cisco</i> definida en la <i>arquitectura de referencia</i> y que permite la administración y configuración del contenido de forma centralizada en un equipo servidor.
DMS	<i>Digital Media System /Digital Media Suite</i> . Solución propietaria de <i>Cisco</i> considerada de referencia para aplicaciones de <i>Difusión Selectiva</i> .
dot1q (Protocolo IEEE 802.1Q)	El protocolo <i>IEEE 802.1Q</i> fue un proyecto del grupo de trabajo 802 de la <i>IEEE</i> para desarrollar un mecanismo que permita a múltiples redes compartir, de forma transparente el mismo medio físico, sin problemas de interferencia entre ellas. Se denomina popularmente como <i>dot1q</i> ("punto 1 Q").
DVB DVB-T DVB-S	<i>Digital Video Broadcasting (Difusión de Video Digital)</i> . Es una organización que promueve estándares aceptados internacionalmente de televisión digital, en especial para televisión de alta definición terrestre (<i>DVB-T</i>) y televisión a través de satélite (<i>DVB-S</i>).

EACCC	<i>Equipo de Administración, Configuración y Carga de Contenido.</i> En el contexto del proyecto es cualquier equipo que disponga de un navegador o agente de usuario web desde el que se realiza la configuración del <i>ESCC</i> (véase <i>ESCC</i>) y/o los <i>ERCs</i> (véase <i>ERC</i>).
ECMA, Ecma International	<i>European Computer Manufacturers Association.</i> Organización internacional cuyo fin es desarrollar, en cooperación con las organizaciones de estándares nacionales, europeas e internacionales, estándares y reportes técnicos con el fin de facilitar y estandarizar el uso de las Tecnologías de Información y Comunicación y Dispositivos Electrónicos. Adquirió el nombre <i>Ecma International</i> en 1994, cuando la <i>European Computer Manufacturers Association</i> (ECMA), cambió su nombre para expresar su alcance internacional.
ERC	<i>Equipo Reproductor Cliente.</i> Es el equipo cliente que se conecta al servidor (véase <i>ESCC</i>) para obtener la configuración y el contenido multimedia y que realiza la reproducción de contenido sobre un medio de presentación (véase <i>SPC</i>). En ocasiones referido como 'equipo terminal'. Es análogo a un <i>STB</i> (véase <i>STB</i>) en un sistema de televisión digital.
ESCC	<i>Equipo Servidor de Configuración y Contenido.</i> Equipo servidor que acepta y gestiona las peticiones de configuración y contenido de los equipos terminales (véase <i>ERCs</i>). Es posible configurar y cargar el contenido que se proporcionará a los <i>ERCs</i> a través de una interfaz gráfica de usuario que permite la administración de estos equipos.
Estándar ECMA	Estándares publicados por <i>Ecma International</i> .
Estándar ISO	Estándares publicados por la <i>Organización Internacional para la Estandarización</i> (véase <i>ISO</i>).
ESVLC	<i>Equipo Servidor VLC.</i> Equipo servidor de contenido que ejecuta la <i>aplicación VLC</i> para tal fin.
firmware	Bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil. El término ha evolucionado para significar casi cualquier contenido programable de un dispositivo de hardware.
Flash Player	Es una aplicación en forma de reproductor multimedia creado inicialmente por <i>Macromedia</i> y actualmente distribuido por <i>Adobe Systems</i> . Permite reproducir, entre otros, archivos y aplicaciones <i>SWF</i> . Tiene soporte para el lenguaje de programación <i>ActionScript</i> , que puede ser usado para mostrar <i>Flash Video</i> (véase <i>FLV</i>) desde un archivo <i>SWF</i> . Debido a que el <i>Flash Player</i> se ejecuta como un <i>plugin</i> del navegador web, es posible incrustar <i>Flash Video</i> y <i>aplicaciones SWF</i> en páginas web y visualizar vídeos por medio del navegador.
FLV	<i>Flash Video.</i> Es un formato contenedor multimedia propietario usado para transmitir vídeo por <i>Internet</i> usando <i>Flash Player</i> . Los contenidos <i>FLV</i> pueden ser incrustados y reproducidos dentro de archivos <i>SWF</i> .
Frame Relay	Técnica de comunicación mediante retransmisión de tramas para redes de circuito virtual.
FTP	<i>File Transfer Protocol (Protocolo de Tránsito de Ficheros).</i>
Gateway	Pasarela, Puerta de Enlace o Router por Defecto.

GPS	<i>Global Positioning System (Sistema de Posicionamiento Global)</i> . Es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un dispositivo.
HTML	<i>HyperText Markup Language (Lenguaje de Marcado de Hipertexto)</i> . Es el lenguaje de marcado predominante para la elaboración de páginas web. Se usa para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
IETF	<i>Internet Engineering Task Force (Grupo Especial sobre Ingeniería de Internet)</i> . Es una organización internacional abierta de normalización que tiene como objetivos el contribuir a la ingeniería de <i>Internet</i> , actuando en diversas áreas, como transporte, encaminamiento, seguridad. La <i>IETF</i> es mundialmente conocida por ser la entidad que regula las propuestas y los estándares de Internet, conocidos como <i>RFC</i> .
IGMP	<i>Internet Group Management Protocol (Protocolo de Gestión de Grupos de Internet)</i> . Protocolo utilizado por los routers que admiten <i>Multicast IPv4</i> , para intercambiar información acerca del estado de pertenencia a grupos <i>Multicast</i> .
ios, IOS	<i>Internetwork Operating System (sistema Operativo de Interconexión de Redes)</i> . Sistema operativo creado por <i>Cisco</i> para programar y mantener equipos de interconexión de redes informáticas como <i>switches</i> (conmutadores) y <i>routers</i> (enrutadores).
interleave	<i>Intercalado</i> . Este concepto es utilizado por muchos contenedores multimedia, de modo que, se refieren a la capacidad del contenedor de intercalar, sobre el mismo archivo, diferentes flujos: vídeo, audio, subtítulos,...
Internet	Conjunto descentralizado de redes de comunicación interconectadas de acceso público.
IPv4	<i>Internet Protocol version 4. Protocolo de Internet versión 4. Es la cuarta versión del protocolo IP y la primera en ser implementada a gran escala.</i>
ISO	<i>International Organization for Standardization (Organización Internacional para la Estandarización)</i> . Es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.
ISP	<i>Internet Service Provider (Proveedor de Servicios de Internet)</i> . Es una empresa que brinda conexión a <i>Internet</i> a sus clientes y usuarios; normalmente es un operador de red.
LAN	<i>Local Area Network (Red de Área Local)</i> . Es un tipo de red de dispositivos cuya extensión está limitada físicamente a un edificio o a un entorno de cientos de metros y que con repetidores podría llegar a la distancia de pocos kilómetros.
LIRC	<i>Linux Infrared Remote Control (Control Remoto Infrarrojo para Linux)</i> .
log	Equivale a la palabra 'bitácora' en castellano. En tecnologías informáticas se refiere al registro oficial de eventos durante un rango de tiempo en particular. Es utilizado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre en un dispositivo o aplicación en particular.

MAC, dirección MAC	<i>Media Access Control (Control de Acceso al Medio)</i> . Es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una Ethernet de red. Es individual y cada dispositivo tiene su propia <i>dirección MAC</i> determinada. Se conoce también como la dirección física en cuanto a identificar dispositivos de red.
máscara de red	Combinación de bits que sirve para delimitar el ámbito de una red de computadoras. Relativo a <i>IPv4</i> .
MBGP	<i>Multicast BGP</i> . Protocolo utilizado entre dominios <i>BGP (Border Gateway Protocol)</i> y utilizado para establecer las tablas de enrutamiento <i>Multicast</i> .
MIME, tipo MIME	<i>Multipurpose Internet Mail Extensions (Extensiones Multipropósito de Correo para Internet)</i> . Son una serie de convenciones o especificaciones dirigidas al intercambio a través de <i>Internet</i> de todo tipo de archivos (texto, audio, vídeo,...) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.
MPEG	<i>Moving Picture Experts Group (Grupo de Expertos en Imágenes Móviles)</i> . Grupo de trabajo del <i>ISO/IEC</i> encargado de desarrollar estándares de codificación de audio y vídeo.
MPLS	<i>Multiprotocol Label Switching (Conmutación de Etiquetas Multiprotocolo)</i> . Mecanismo de transporte de datos estándar creado por la <i>IETF</i> . Trabaja entre la capa de enlace de datos y la capa de red del <i>modelo OSI</i> . Fue diseñado para unificar el servicio de transporte de datos para las redes basadas en circuitos y las basadas en paquetes.
NAT, red Nateada	<i>Network Address Translation (Traducción de Dirección de Red)</i> . Es un mecanismo utilizado por <i>routers IPv4</i> para intercambiar paquetes entre dos redes que han asignado direcciones privadas. Consiste en convertir en tiempo real las direcciones privadas, utilizadas en los paquetes transportados, en direcciones públicas. Existen diversos métodos para aplicar NAT. Una <i>red Nateada</i> será aquella cuyos equipos con <i>direcciones IPv4</i> privadas accedan a <i>Internet</i> con una <i>dirección IPv4</i> pública compartida por varios equipos.
NTP	<i>Network Time Protocol (Protocolo de Hora de Red)</i> . Es un protocolo diseñado para sincronizar los relojes de los sistemas informáticos a través del encaminamiento de paquetes en redes con latencia variable.
OSI	<i>Open System Interconnection (Modelo de Interconexión de Sistemas Abiertos)</i> . Propuesta que hizo la Organización Internacional para la Estandarización (ISO) para estandarizar la interconexión de sistemas abiertos.
OSI, modelo OSI	<i>Open System Interconnection (Interconexión de Sistemas Abiertos)</i> . El <i>modelo OSI</i> es el modelo de red descriptivo creado por la <i>Organización Internacional para la Estandarización</i> (véase ISO) en el año 1984. Es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.
PIM	<i>Protocol Independent Multicast (Protocolo Multicast Independiente)</i> . Protocolo que crea una estructura de árbol de distribución entre los clientes multicast, de forma que se establecen dominios.

plugin	También se lo conoce como <i>plug-in</i> , <i>add-on</i> , <i>complemento</i> , <i>conector</i> o <i>extensión</i> . Es una aplicación que se relaciona con otra, para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de una <i>API</i> .
Proxy	Es un programa o dispositivo que realiza una acción en representación de otro.
Proxy-Caché, (Arquitectura)	Es el equipo que realiza las funciones de <i>proxy</i> en una <i>LAN</i> con respecto al <i>ESCC</i> . Se ha denominado <i>arquitectura Proxy-Caché</i> a la utilización de un <i>ERC</i> como <i>proxy</i> con respecto al <i>ESCC</i> del contenido multimedia hacia el resto de <i>ERCs</i> de la misma <i>LAN</i> .
RAM, memoria RAM	<i>Random-Access Memory (Memoria de Acceso Aleatorio)</i> .
RFC	<i>Request For Comments (Petición de Comentarios)</i> . Documentos publicados por el <i>IETF</i> . Cada RFC es un documento cuyo contenido es una propuesta oficial para un nuevo protocolo de la red <i>Internet</i> . Antes de que un documento tenga la consideración de <i>RFC</i> , debe seguir un proceso muy estricto para asegurar su calidad y coherencia.
RSTP	<i>Real Time Streaming Protocol (Protocolo de Flujo de Datos en Tiempo Real)</i>
RTCP	<i>RTP Control Protocol (Protocolo de Control RTP)</i> . Es un protocolo de comunicación que proporciona información de control. Esta información está asociada con un los datos de un flujo RTP.
RTMP	<i>Real Time Messaging Protocol (Protocolo de Mensajes en Tiempo Real)</i> . Protocolo desarrollado por Adobe para la transmisión de contenido multimedia en tiempo real a modo de Streaming. Tiene diferentes variantes: <i>RTMTP</i> (véase), <i>RTMPS</i> (véase <i>RTMPS</i>) y otros.
RTMPS	<i>RTMP with Secure Tunnel (RTMP con Túnel Seguro)</i> . Es una variante de <i>RTMP</i> que se implementa sobre <i>HTTPS</i> .
RTMTP	<i>RTMP with Tunnel. (RTMP con Túnel)</i> . El protocolo funciona de forma similar a <i>RTMP</i> pero se establece un túnel mediante <i>HTTP</i> con la intención de evitar problema por la configuración de los firewalls de la red.
RTP	<i>Real-time Transport Protocol (Protocolo de Transporte en Tiempo Real)</i> . Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real.
SFTP	<i>Secura Fila Transfer Protocol (Protocolo de Transferencia de Fichero Seguro)</i> . Es un protocolo de red que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos sobre un flujo de datos fiable.
SPAN	<i>Switched Port Analyzer (Analizador de Puertos Conmutados)</i> .
SRTCP	<i>Secura RTP Control Protocol (Protocolo de Control RTP Seguro)</i> . Protocolo que añade seguridad a <i>RTCP</i> .
SRTTP	<i>Secura Real-time Transport Protocol (Protocolo de Transporte en Tiempo Real Seguro)</i> . Protocolo que añade seguridad al protocolo <i>RTP</i> .

STB	<i>Set-top Box</i> . Es el nombre con el que se conoce el dispositivo encargado de la recepción y decodificación de señal de televisión analógica o digital (DTV: cable, satélite, terrestre, IPTV,...), para luego ser mostrada en un dispositivo de televisión.
STD	Estándares de <i>Internet</i> documentados y publicados por <i>Internet Engineering Task Force</i> (véase <i>IETF</i>).
SWF	<i>Small Web Format (Formato de Web Pequeño)</i> (inicialmente abreviación <i>Shockwave Flash</i>). Es un formato de archivo de gráficos vectoriales creado por la empresa <i>Macromedia</i> (actualmente <i>Adobe Systems</i>). Los archivos SWF pueden ser creados por el programa <i>Adobe Flash</i> aunque hay otras aplicaciones que también lo permiten. Suele ser ejecutado sobre el navegador mediante un <i>plugin</i> llamado <i>Adobe Flash Player</i> , aunque también pueden ser encapsulados para ejecutarse de forma autónoma. Básicamente es un formato vectorial aunque también admite <i>bitmaps</i> , con posibilidades de animación. También admite programación mediante el lenguaje <i>ActionScript</i> .
syslog	Sistema de registro de mensajes de un sistema informático.
TCP	<i>Transmission Control Protocol (Protocolo de Control de Transmisión)</i> .
TFT	<i>Thin-Film Transistor (Transistor de Película Fina)</i> . Una de las aplicaciones del material TFTs son las pantallas de cristal líquido. Comúnmente se denomina TFT, a las pantallas construidas con este material.
TFTP	<i>Trivial File Transfer Protocol (Protocolo de Transferencia de Archivos Trivial)</i> .
udev	Es el gestor de dispositivos que usa el <i>kernel Linux</i> en su versión 2.6. Su función es controlar los ficheros de dispositivo en <i>/dev</i> . Maneja el directorio <i>/dev</i> y todas las acciones del espacio de usuario al agregar o quitar dispositivos, incluyendo la carga de firmwares.
UDP	<i>User Datagram Protocol (Protocolo de Datagramas de Usuario)</i> .
UHF	<i>Ultra High Frequency (Frecuencia Ultra Alta)</i> . Es una banda del espectro electromagnético que ocupa el rango de frecuencias de 300 MHz a 3 GHz. Uno de los servicios UHF más conocidos por el público son los canales de televisión, tanto locales como nacionales. Actualmente se usa la banda UHF para emitir la Televisión Digital Terrestre.
URI	<i>Uniform Resource Identifier (Identificador de Recurso Uniforme)</i> . Similar al <i>URL</i> pero se incluye el <i>fragmento</i> , además de la <i>parte jerárquica</i> y la <i>solicitud o consulta</i> del recurso en cuestión.
URL	<i>Uniform Resource Locator (Localizador de Recursos Uniforme)</i> . Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en <i>Internet</i> para su localización o identificación. El concepto de <i>URL</i> ha sido incorporado dentro de uno más general (véase <i>URI</i>).
W3C	<i>World Wide Web Consortium</i> . Abreviado <i>W3C</i> . Es un consorcio internacional que produce recomendaciones para la <i>World Wide Web</i> .
WAE	<i>Wide Area Application Engine (Motor de Aplicaciones de Área Amplia)</i> .
WAN	<i>Wide Area Network (Red de Área Amplia)</i> . Es un tipo de red de dispositivos

capaz de cubrir distancias desde unos 100 hasta unos 1000 km.

Wi-Fi	Tecnología de redes inalámbricas de área local que adopta los estándares 802.11.
WiMAX	<i>Worldwide Interoperability for Microwave Access (Interoperabilidad mundial para Acceso por Microondas)</i> . Tecnología de transmisión de datos que utiliza las ondas de radio en las frecuencias de 2,3 a 3,5 Ghz. El protocolo que caracteriza esta tecnología es el <i>IEEE 802.16</i> . Una de sus ventajas es dar servicios de banda ancha en zonas donde el despliegue de cable o fibra, por la baja densidad de población, presenta unos costos por usuario muy elevados (zonas rurales).
WPAN	<i>Wireless Personal Area Networks (Red de Área Personal Inalámbrica)</i> . Es una red para la comunicación entre distintos dispositivos cercanos (pocos metros) al punto de acceso.
WWW, World Wide Web	WWW. Sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de <i>Internet</i> .
XHMTL	<i>eXtensible Hypertext Markup Language (Lenguaje eXtensible de Marcado de Hipertexto)</i> . Es el lenguaje de marcado pensado para sustituir a <i>HTML</i> como estándar para las páginas web y que cumple ciertas especificaciones más estrictas de <i>XML</i> .
XML	<i>eXtensible Markup Language (Lenguaje de Marcas eXtensible)</i> . Es un metalenguaje extensible de etiquetas desarrollado por el W3C. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.